LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

Feasibility of Cloud Services for Customer Relationship Management in Small Business

A Drupal Approach towards Software as a Service

LAHTI UNIVERSITY OF APPLIED SCIENCES
Degree programme in Business Studies
Business Information Technology
Thesis
Winter 2012
Liang Sun

Lahti University of Applied Sciences
Degree Programme in Business Studies

SUN, LIANG: Feasibility of Cloud Services for Customer Relationship Management in Small Business
A Drupal Approach towards Software as a Service

Bachelor's Thesis in Business Information Technology, 56 pages, 16 pages of appendices

Winter 2012

ABSTRACT

Information has value. All kinds of business should have information systems for business growth. Software is used for business management and entrepreneurship resource planning worldwide. However, its shortages turn out to be more and more obvious in recent years. Software as a Service (SaaS) is thus regarded as one serious alternative of software. Customer Relationship Management (CRM) is of vital importance in today's business. SaaS adoption for CRM then becomes a trend in Small and Medium Businesses. Home delivery business operation in a small restaurant calls for such adoption.

The purpose of this research was to demonstrate a CRM-oriented SaaS prototype for home delivery business operation in a small restaurant. Another aim was to report the feasibility of SaaS for CRM in this restaurant business operation. Finally, the question that what kind of SaaS application for CRM might work in such business operation was answered in this study.

This thesis was carried out all the way through Design Science Research Approach in an inductive way. Data was collected through experimental Drupal implementation of the Home Delivery Customer Relationship Management system (HDCRM) and then analyzed against home delivery business workflows and time spent for each activity in real life.

This study discovered that Software as a Service is adoptable for Small & Medium Businesses. The results also indicated that home delivery business demands Customer Relationship Management system. The principal conclusion was that the HDCRM demonstrated in this paper meets home delivery business needs.

Keywords: Software as a Service, Customer Relationship Management, Drupal, Home Delivery Business

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Information has value. Avison & Fitzgerald (2006) thus conclude that all kinds of organizations have information systems. Compared to manual systems within which information is considered timely of value, computer-based information systems can process data much faster and more accurately as well as provide data when and where required at a more correct level of detail. (Avison & Fitzgerald, 2006) Computer-based information systems spread extensively, and have been and are being used widely for management work in many industrial sectors. On the other hand, traditional manual systems have been redefined simultaneously in different fields.

Software used to be and is still a fundamental element of computer-based information systems. It is now everywhere in the developed world. (Tucker, et al., 2011) Software applications designed for business management and entrepreneurship resource planning (ERP) are used for improving efficiency and enhancing effectiveness worldwide. These application services are typically installed into local disks by end users and a certain amount of licensing costs would need to be paid to Application Services Providers (ASP) who develop those particular applications.

Although software applications are rich enough to be used everywhere, software, as one mode of computer-based information systems, has shortages. From perspective of software project managers, one of these shortages is that software projects are commonly late and over budget so as to cause the product to be late to market (Blackburn & Scudder, 1996). One other drawback, from the end users' view, can be lack of mobility and too much hardware and licensing costs.

Internet has brought the whole world to a new era of information systems. Stephen Todd notes in his research paper "A guide to the Internet and World Wide Web" that the Internet is a set of interconnected networks that may themselves contain other networks or just computers around the world. With HTTP protocol, World Wide Web then introduces HyperText Maskup Language to web development , which allows web content to be connected with hyperlinks. (Todd, 1999). This whole developing process triggers an increasing

amount of Web site development work being carried out in various organisations. (Taylor, et al., 2001)

Thereafter, the dot com bubble in the 1990s changed many things in the Internet community, some projects failed completely, but many others remained stable and profitable by sacrificing loss of a large portion of their market capitalization (2012). Afterwards, the way of delivering those applications or, let's say it more accurately, services in the digital world starts to shift from locally installed software and packages to cloud-based applications that can be accessed easily through a web browser or other forms such as mobile apps. With regard to these, cloud-based applications are more like services than products to end users, they are well known as "Software as a Service". (Software & Information Industry Association, 2001) It is believed SaaS would overcome those shortages software has.

Customer Relationship Management has a variety of meanings (Buttle, 2009). Kotler (2010) concludes that CRM normally refers to those activities and processes of building and maintaining profitable relationships by creating and delivering value and satisfaction to customers. Thus, relations of current customers are retained and that of new customers are created.

There was continuous CRM development by using information technologies for several decades. Understanding and emphasis on different aspects of CRM has determined disparate information technology to be used. Trace back to the 1970s or earlier, there were a series of technologies that including call centers, sales-force automation (SFA) systems and customer information files (CIF). (Buttle, 2009) Thereafter, web technologies started to be a significant role in such attempt to help users access and share information without having software application installed locally. SIIA defines CRM system as "*an integrated information system that is used to plan, schedule and control the presales and post-sales activities in an organization*" (Software & Information Industry Association, 2001).

Salesforce (2012) brings the world a well designed and fully featured CRM SaaS application called "Salesforce.com for CRM", however, Sales Salesforce.com for

CRM is targeted at large enterprises rather than Small & Medium Businesses (SMBs), there is rare consideration of web development for small companies or oraganizations.

Throughout a CRM-oriented SaaS prototype for Home Delivery Customer Relationship Management (HDCRM), this paper aims to report the feasibility of SaaS for CRM in home delivery business operation in a small restaurant. Moreover, it tries to answer the following research question:

1. What kind of SaaS application for CRM might work in home delivery business operation in a small restaurant?

The research method chosen for this thesis is Design Science Research Approach. Besides, this thesis is conducted as an inductive and experimental research. It intends to illustrate possibilities and potentials of Software as a Service for CRM in SMBs by giving a Drupal (Version 7.16) demonstration for Customer Relationship Management in home delivery business operation in a small restaurant.

In this thesis, there are five general parts of study. Firstly, a brief knowledge base about concepts such as cloud computing and SaaS and the situation of SaaS in CRM is given. Secondly, the author presents Design Science Research method used for this paper. Thirdly, by going through a systems development life cycle, the HDCRM is presented and data is correspondingly collected during the whole process. Furthermore, the author contributes analysis of data collected through application development comparing with real life business workflows and activities. At last, Conclusions Section summaries results of the study as well as limitation of research scope. Additionally, the author tries to give several recommendations for further research.

2 LITERATURE REVIEW

According to International Telecommunication Union (ITU)'s statistics, one third of the world's population was using the Internet by the year of 2011 (2011). As Internet and World Wide Web grow up, major difference has taken place among the way computer-based information systems behave. According to statistics and predictions from Gartner, cloud computing is getting matured and more than half of Global 1000 companies will move customer-sensitive data to the public cloud by the end of year 2016 (Pettey, 2011). It can be then anticipated that cloud computing is going to be the future for both service providers and service end users. Salesforce thus firmly states in one of their whitepapers that,

> *"…for service customers, the benefits are obvious and compelling that they would get sophisticated functionality without any up-front expenses or hassles related to installation and maintenance of locally installed software applications; for suppliers of such services, the model provides low barriers to entry and unprecedented opportunities as well as new risks and challenges…". (Salesforce.com, 2008).*

In this section, there are three main subsections. In the first subsection, the author introduces basic background information about Cloud Computing and Software as a Service. Then, the reasons for using Drupal in this study are presented. At last, the author summarizes this whole chapter.

2.1 Cloud Computing and SaaS

In this subsection, basic concepts such as cloud computing and SaaS is detailed discussed. Moreover, development circumstances of SaaS for CRM in recent years are presented.

2.1.1 Cloud Computing

Cloud computing has been a popular buzzword in digital world in recent years. Cloud computing usually refers to on demand computing, a new way of computing by which a service rather than a product is delivered to end users and is paid per usage or through a periodical subscription (Wikipedia, 2012). Meanwhile, resources and information are transferred among computers and other

devices as a utility over the Internet (Armbrust, et al., 2010). People in computer science field have dreamed of this for many years. Today, this long held dream is becoming a reality (Smith, et al., 2011). However, the debate about the characteristics, or the nature, of cloud computing is widely arisen. Attention is paid also to the essential differences between cloud computing and traditional way of computing, such as client-server model and utility computing (Armbrust, et al., 2010). Many people believes that cloud computing is changing the way we do everything in the world, while others think cloud computing is just another fancy way for technical companies doing marketing (Kepes, 2010).

Despite the discussion, cloud computing is gaining exposure. It should have the potential to end the battle between licensing and open source, simply because of the subscription-like pricing. Moreover, the new model of handling information tends to be the mainstream of this age, changing the world as Internet used to.

Cloud computing is categorized in three different levels of service model. Sharing the same characteristics with utility computing, the so-called Infrastructure as a Service (IaaS) gives consumers an option to rent computational resources such as computation and storage rather than acquire them, so that the initial costs can be lower or even zero. The second service model of cloud computing is Platform as a Service (PaaS). A computing system or a solution stack is provided as a service through PaaS. Microsoft Windows Azure and Amazon Web Services are two, among others, of this kind of services. The third service model of cloud computing is Software as a Service. (Chee & Franklin, 2010)

2.1.2 Software as a Service

SaaS is well known as an “on demand” software application using web services due to its typical pricing model. End users obtain access or permission by paying a monthly or annual fixed fee while the actual data is hosted by the service provider centrally (Wikipedia, 2012). Criticisms argue that SaaS is just a new term of online software web services, since SaaS applications are essentially web applications. Yet there are lots of researches indicate and support the viewpoint that SaaS has such a number of features that differentiate SaaS applications from

other software applications (Box, 2012). Three most important ones of these characteristics are mobility, elasticity and configurability.

2.1.2.1 Mobility

Mobility indicates that access to information systems is much wider and easier. With all data stored into the cloud, Software as a Service can be reached anywhere at any time. Mobility has become a top priority for IT and business leaders in all kinds of organizations, most popular social networking sites such as Facebook, Twitter, and Foursquare are already SaaS-based. Since mobility is being driven more and more by consumerization, it is highly likely that enterprise mobility applications and management tools will also be delivered in the same way. (MaaS360, 2011)

2.1.2.2 Elasticity

Elasticity refers to the quick scalability and flexibility of the data model. The Software & Information Industry Association (SIIA) states in one of its whitepapers that rapid elasticity is one, among others, benefit and feature that driving cloud computing to be quickly adopted. As an economic impact, it then boosts other economic activity via the provision of more effective and less expensive computing capabilities. (Software & Information Industry Association, 2011)

2.1.2.3 Configurability

Configurability means SaaS has sufficient ability for application customization, that is, with the same codebase and instance of application running on the server end, there can be multiple tenants subscribed into it, share it, and for different clients this kind of mature SaaS offering would have ability to modify preferences. Due to the underlying application codebase cannot be changed for each tenant's customizations, the application is abstracted into loosely coupled fine-grained configurable components. Consequently, customers can make quick change the look and feel, logic, and database layers of the application through

on-screen clicks without code modification, compilation, or deployment. (Hai & Sakoda, 2009)

2.1.3 SaaS for CRM in SMBs

Many IT providers consider SaaS as a serious alternative to locally installed software. Amongst these vendors, there are hundreds of them endeavored to push SaaS adoption in CRM field. Benlian and two other researchers illustrate their analysis of different types of SaaS-adoption levels in 2008 and 2010 for both Small and Medium Businesses and large enterprises with FIGURE 1. (Benlian, et al., 2009)

	Adoption levels of SMBs (% of respective IT application budget)		**Adoption levels of large enterprises (% of respective IT application budget)**	
Application type	2008	2010	2008	2010
Collaboration	10 %	35 %	13 %	43 %
Content Management	6 %	25 %	17 %	36 %
Office	6 %	23 %	9 %	28 %
CRM	5 %	32 %	15 %	37 %
Human Relations	3 %	19 %	7 %	23 %
Business Intelligence	3 %	17 %	1 %	11 %
ERP	3 %	10 %	1 %	10 %
SCM	0 %	2 %	0 %	1 %
Production	0 %	0 %	3 %	4 %
Engineering	0 %	3 %	0 %	4 %
Total average	5 %	21 %	6 %	23 %

FIGURE 1. SaaS adoption levels of SMBs and large enterprises (2009)

In the tabular figure above, it is quite noticeable that IT application budget spent on CRM toward SaaS adoption in SMBs would have increased more than six times by 2010 as that in the year of 2008. Within SaaS adoption that driven by

these IT suppliers, Salesforce for CRM is regarded as one of the most successful SaaS application in CRM market (Pring, 2005).

Salesforce.com was ranked number 27 in Fortune's 100 best companies to work for in 2012 with its "social enterprise" strategies. However, its pronouncement also limits its goals and targets to providing services commonly for large enterprises. (Wikipedia, 2012)

It is widely accepted that there are plenty of potentials and possibilities in SaaS adoption for CRM in SMBs. Nowadays, more and more SMBs are involved in this revolution (Gold, et al., 2004).

2.2 Why Drupal?

2.2.1 Web Development Concern

Software as a Service development, according to what devices to be used as endpoint access devices, can be divided into Desktop/Laptop oriented SaaS development that is essentially web services development, and Mobile SaaS development. When referred to Desktop/Laptop oriented SaaS, the author means to refer to that the services are accessed and used throughout a web browser. Similarly, Mobile SaaS are deployed as a mobile app in mobile devices.

Due to the limitation of the scope of this paper, Drupal Content Management Framework (CMF) is chosen for the SaaS application development. That is, this paper will concentrate on using Drupal to develop a web oriented SaaS application that would suit the context of home delivery business operation in a small restaurant, even though some methodologies and tools used during system analysis and design can and will be used for other types of development patterns, mobile services for example, in the future.

2.2.2 Open Source Concern

The second reason for choosing Drupal is that it is an open source content management system. The main concern moving to the cloud is to reduce

unnecessary costs. Using a free Open Source CMF as the back-end framework benefits both IT project managers and service end users.

2.2.3 Programming Concern

Comparing to other popular free open source Content Management Systems (CMS) such as Joomla! and Wordpress, Drupal is relatively developer-friendly. Besides, it is more powerful to scale up massive distributed websites, which perfectly suits the requirements SaaS and cloud computing ask for.

2.3 Conclusion

In this chapter, major trends in Cloud Computing and SaaS are presented. It is believed Cloud Computing is the way many IT vendors want to provide their products. Some researchers even start to think of Everything as a Service (XaaS) (Duan, 2012).

SaaS for CRM is such a considerable topic for IT developers. There are many IT vendors who endeavor in pushing SaaS adoption in CRM field these days. For several concerns, Drupal is chosen as main Content Management Framework to deliver a local host application for CRM in home delivery business operation in a small restaurant.

The next chapter will present a real life case so as to illustrate problems existing in home delivery business operation when concerning about Customer Relationship Management.

3 CASE PRESENTATION

Practice makes perfect. Based on the author's part-time working experience in a restaurant, observation of problems that exist in restaurant business operation has been accumulated. Although other parts of business operation such as human resource management, material stock management and financial management are of vital importance in restaurant management, this paper will focus strictly on Customer Relationship Management and enhance of such management work by using SaaS application services.

As a witness of the whole process of establishing home delivery business and the main executive of the food delivery work flow in a relatively long period, the author himself particularly cares about issues of making this work easier. From a view of ICT, these issues include:

1. Transforming manual paper system to computer-based information system;
2. Storing customer information into computer database;
3. Retrieving database information anytime anywhere;
4. Conveying delivery information via Internet.

Home delivery business operation is more about Customer Relationship Management than any other selling activities occuring inside restaurant base in restaurant businesses. This is because customer information is of great concern in home delivery business operation for connection between the two ends, customers and restaurants. Ordering and consuming food inside a restaurant requires no information, for instance, a home address to be left to the restaurant. However, such customer information is crucial for food home delivery. For this reason, a series of problems has been identified as follows:

1. Applicability concern:
    a. Some customer information can be so important to customers that some of them may not be willing to leave it;
    b. If customers doesn't want to leave much information, how to still make business connections;
2. Security concern:

a. If customers trust the restaurant and give enough information, how to protect these customer information;

The importance of working out these problems so as to improve restaurant management level is therefore highlighted. These problems lead to a result of careful selection of customer information. What to take and what not to take is then ultimate solution to take great care of customer information and manage business connections between sellers and buyers. Home delivery business flow is very simple. Buyers need to call sellers for ordering and sellers might have to call buyers when buyers cannot be reached for some reasons. In addition, location addresses need to be determined and agreed so that food can be delivered and business deals can be accomplished. Hence, phone number and location address would be two musts when collecting customer information.

Thus, for Issue 1) mentioned in the previous page, when converting manual paper system to computer-based system, phone number and location address are two required fields to be taken and recorded while others can be optional. As a successful migration to computer-based system, the three following issues mentioned in the previous page are just a matter of system architecture of such information system.

4 RESEARCH APPROACH

The research method chosen for this thesis is Design Science Research approach. Alan R Herver said in his article, “Design Science in Information Systems Research”, the Design Science Research method “*seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts*” (Hevner, et al., 2004). Herver thereafter gave seven guidelines for researchers to use this method. This paper demonstrates a viable SaaS artifact to provide a suitable solution for small companies and organizations, a restaurant in this case. The following text aims to give detailed description of how Herver’s seven guidelines apply in this paper.

4.1 Design as an Artifact

During research of this thesis, a purposeful artifact, the SaaS for managing home delivery business operation, is demonstrated. Not only constructs and models but also instantiation is given to evaluate the applicability of Software as a Service adopted into small business. In order to decrease cost of application development, concentration on usability and secure data flow for each participant involved into home delivery business operation is what this paper is mainly about.

4.2 Problem Relevance

As mentioned previously, locally installed softwares and packages have several disadvantages such as less mobility and overdue deployment. These weaknesses become more and more obvious after the rise of cloud computing and Web2.0. The SaaS application prototype in this document aims to address and try to solve similar issue. Home delivery business operation is handled by paper-based system in the past, and a number of problems are derived from this old system. Such problems include:

1. Seller has to ask buyers’ general information over and over again, even if buyers have ordered before but valuable information is ignored as those papers for recording have been thrown as rubbish, or it becomes a difficult task to maintain as they accumulate more and more.

2. Knowledge management is well known as an important part of business operation. However, mistakes cannot be avoided occasionally. When Waiter/Waitress makes mistakes especially about location addresses, workflow would be hindered and resources such as time and money are therefore wasted. Likewise, new workers involved know nothing about previous ordering information and it becomes really hard for them to manage their work properly.
3. Information communication is as crucial as knowledge management during business operation. In home delivery business operation there are typically three restaurant workforces involved: Reception Desk, Manger/Boss and Delivery Person (Kitchen Worker is skipped in this paper due to limitation of the research scope). For the last participant, general information has to be given in order to make his/her work more fluent. Example of such information might be new orders coming in while transporter is still driving on the way.

4.3 Design Evaluation

In essence, the functionality, completeness, consistency, accuracy, performance, reliability and usability of this SaaS application would define the overall design evaluation of itself. Specifically, this research will study the applicability of the SaaS application for a purpose of effective and efficient way of working. To achieve a good result of design evaluation, modeling and design of the application should meet business needs in home delivery business operation.

4.4 Research Contribution

Firstly, the designed artifact would be capable of solving the problems addressed. By identifying and finally fulfilling business needs of home delivery business operation, the instantiation of the designed artifact would increase efficiency and effectiveness in home delivery business operation in a small restaurant. Secondly, the knowledge base would be extended and improved due to the work of this research. Research problem of this paper is concentration on the applicability of the designed artifact in home delivery business operation by framing the

designing process and evaluating the designed artifact itself, consequently, knowledge management and information communication is enhanced. At last, this research would prove the implementability of similar applications for companies and organizations. Although this paper is primarily about home delivery business operation in restaurants, it is believed that Software as a Service could be equally applicable for other types of business operations.

4.5 Research Rigor

In this research, after the construction of the designed SaaS application, data is collected throughout the whole process of experimental development processes. Additionally, the author would gather workflow data and time spent for each activity from a restaurant where paper manual system is still used as main driver of home delivery business operation. The designed artifacts and the designing process would evaluate the applicability of Software as a Service in home delivery business operation itself via evaluation components such as functionality, consistency, performance and usability.

4.6 Design as a Research Process

Requirement specification is generated on account of the research problems and business needs in real working life at the very beginning. Therefore, the SaaS application instantiation is created through a series of application development activities according to the requirement specification. Afterwards, the instantiation is tested against the requirement specification so as to meet business needs. For evaluation of the applicability of Software as a Service in home delivery business operation, this process is therefore executed iteratively.

4.7 Communication of Research

This SaaS application is generated specifically for customer relationship management for home delivery business operation in small restaurants. Thus, the implementation is supposed to be suitably used within the organizational context

and the organizational resources should be committed to use this application within their specific organizational context.

4.8 Conclusion

All in all, this paper is intended to give a SaaS artifact prototype that would solve relevant problems addressed, with obligations to evaluate itself against certain arguments iteratively and finally achieves the research goal declared in this thesis.

5 DATA COLLECTION

To collect adequate data for this research, this paper will follow instructions given as information systems development life cycle (SDLC) that are presented in Avison & Fitzgerald's book "Information Systems Development: methodologies, techniques & tools" (2006). This waterfall model consists of six stages in systems development process:

1. Feasibility study;
2. System investigation;
3. Systems analysis;
4. Systems design;
5. Implementation;
6. Review and maintenance.

The following text tries to demonstrate the HDCRM artifact in a Drupal approach according to these six stages.

5.1 Feasibility Study

As indicated in the previous chapter, traditional manual paper system has difficulties in information recording, data control, knowledge management and information communication. Computer-based system can however handle all these difficulties appropriately and furthermore provide a number of new possibilities. Some of these possibilities would be:

1. Storing customer information in one single database to be easily retrieved for Customer Relationship Management;
2. Maximizing the potential value of customer information by enhancing information communication;
3. Allowing integration of other external systems with present management system. For instance, an E-commercial system can be thereafter connected with the HDCRM;

Besides new possibilities, computer-based system puts more emphasis on benefits and practical applicability that meet present business needs.

5.2 System Investigation

To meet restaurant business needs in home delivery business operation, the System Investigation stage presents functionality, usability and security requirements obtained mainly throughout observations. Before investigating these requirements, it is necessary to introduce people involved into home delivery business operation. They are Waiter/Waitress (W in text below), Delivery Person (D), Manager/Boss (M) and Customer (C).

5.2.1 Functionality Requirements

Functionality defines how information system fulfills each of its purposes. In home delivery business operation, it is concluded that several functionality requirements are necessary as follows:

1. When W receives a phone call, he/she needs to record all information C gives, including required fields such as phone number and location address, optional fields such as customer name, food kinds, ordering time and payment methods;
2. When W receives a phone call, he/she would be able to search existing customer information with one particular field given, for instance, phone number;
3. When W made an order, he/she inform D ordering information via Internet;
4. When D received new order notification, he/she would be able to access order receipts via Internet, mobile application is much better if possible, so as to proceed food delivery to C;
5. When D check location address of C, a graphical map like Google maps would be presented;
6. M would be able to check overall order statistics, for example the amount of sales in a day or a certain period;
7. M could also view a full map presentation of sales in a day or a certain period;
8. M may check the amount of sales according to certain condition filter such as payment methods or single particular customer;

5.2.2 Usability Requirements

For better user experience, usability requires generally friendly user interface to be presented. Above all, a system has to be operational by a specified user performing a specified task in a specified environment. According to document DIS 9241-11 from the International Organization for Standardization (ISO), Faulkner thus concludes there are four usability attributes among others (Faulkner, 2000). These four attributes are effectiveness, efficiency, user satisfaction and learnability.

5.2.2.1 Effectiveness

Firstly, by disassembling the whole development process of the HDCRM system into several sub jobs, the success of failure ratio in this work is achieved. Secondly, localization features are added to enhance effectiveness of the HDCRM system. Finally, output given out by certain functionality such as search filter would improve effectiveness of the system.

5.2.2.2 Efficiency

The HDCRM system is designed to reduce time used for performing certain tasks in home delivery business operation. However, time spent for learning to use the system can never be dramatically reduced.

5.2.2.3 User satisfaction

It is unlikely to measure how users feel about using the system. However, the author supposes that by means of a well-designed artifact with sufficient observation of problems existed and system investigation, this goal can be achieved.

5.2.2.4 Learnability

Faulkner again states in her book, "*A system should be easy for the user to learn so that it is possible to use the system effectively as quickly as possible*". It is believed

a system is of easy learnability while a novice user can easily perform tasks by using such a system. This is certainly what the HDCRM system intended to achieve.

5.2.3 Security Requirements

Data security is and has to be an important concern for both cloud service providers and users. Most of them have certain expectations for security of their data. Winkler (2011) mentioned in his book that "*A good starting point when you need to measure the presence and effectiveness of the security of a cloud includes having a list of required or recommended security controls*". There are a lot to be referred about security controls on infrastructure and platform level. However, on application level, such security control would be limited only to user roles access permissions. In this paper, the HDCRM system is demonstrated through a Drupal approach, and Drupal states in its Content Management System: "*Permissions let you control what users can do and see on your site. You can define a specific set of permissions for each role*". Thus, it is concluded that the HDCRM system, which is developed through Drupal Content Management Framework, is of great security controls on application level.

5.3 Systems Analysis

Systems analysis is all about endeavors to understand every aspects of the system. In order to specify, visualize and construct the HDCRM system in an object oriented analysis way, the Unified Modeling Language (UML) for modeling is chosen. UML provides

> *...system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.*

In this section, there illustrate a class diagram for structures, a use case diagram for behaviors and a sequence diagram for extending behaviors to interactions.

5.3.1 Class Diagram

A class diagram in the UML is a type of static structure diagram that shows the objects of the system, their attributes, operations, and the relationships among the objects.

To formulate a class diagram in home delivery business operation, objects that are considered important are Waiter/Waitress (W), Order (O), Customer (C) and Delivery person (D).

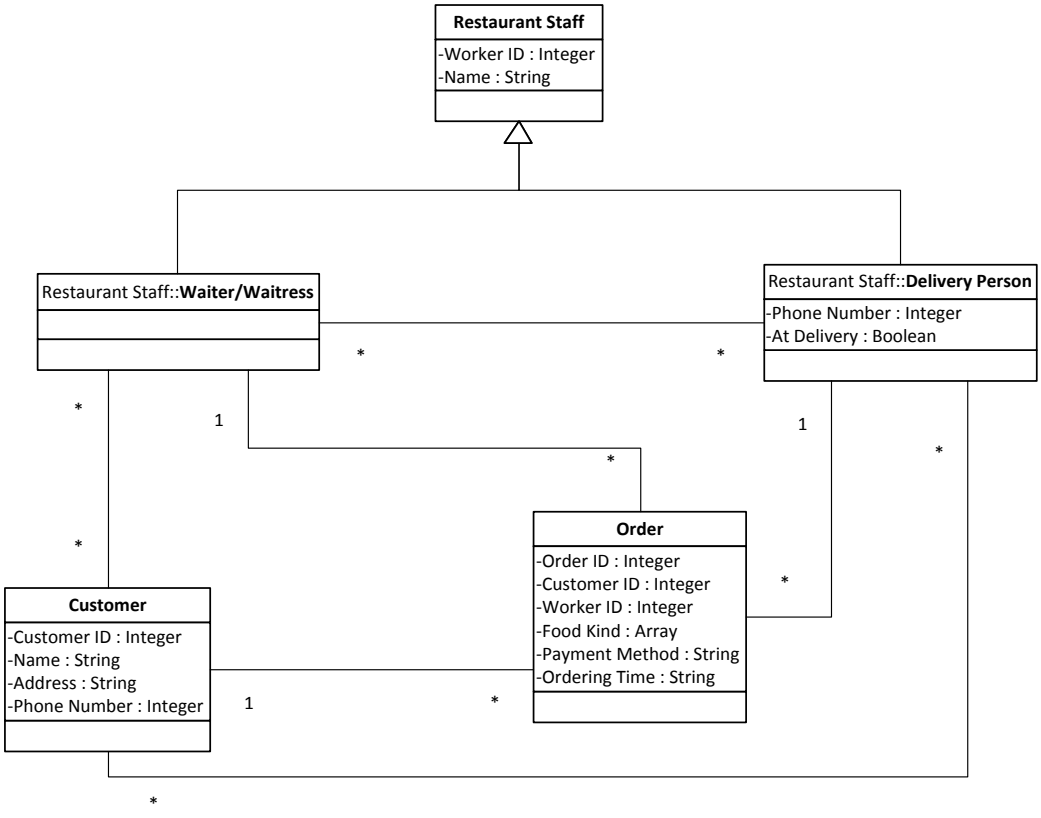

FIGURE 2. Class Diagram for HDCRM.

Notice that in FIGURE 2, it presents five classes within which four of them are of vital importance, while Class Restaurant Staff is only performed as generalization

of Waiter/Waitress and Delivery Person class. Each of these five classes has a primary key, while Class Order has also foreign keys to refer all attributes of class Customer and that of class Waiter/Waitress or Delivery Person. The relationships between each other are as follows:

1. Class Restaurant Stuff is composed of Class Waiter/Waitress and Delivery Person;
2. One waiter/waitress would link to many delivery person, and one delivery person may talk to different shift-working waiters/waitresses;
3. One waiter/waitress may receive phone call from many customers, and one customer may order from different shift-working waiters/waitresses;
4. One delivery person may delivery food to many customers, while one customer may get delivery from different delivery person as well;
5. One customer may make many orders, but one order is made by only one customer;
6. Similarly, one waiter/waitress may record many orders, but one order is recorded by one single waiter/waitress;
7. Also, one delivery person may ship many orders, but one order is shipped by one particular delivery person.

5.3.2 Use Case Diagram

A Use Case diagram is a graphical illustration of each pieces of functionality the system provides and each user involved into the system. It tells what the system will do and who will interact with it. Functionalities of the HDCRM system for home delivery business operation have been identified in Functionality requirements Section above. The Use Case diagram shown in next page visualizes these functionality requirements.

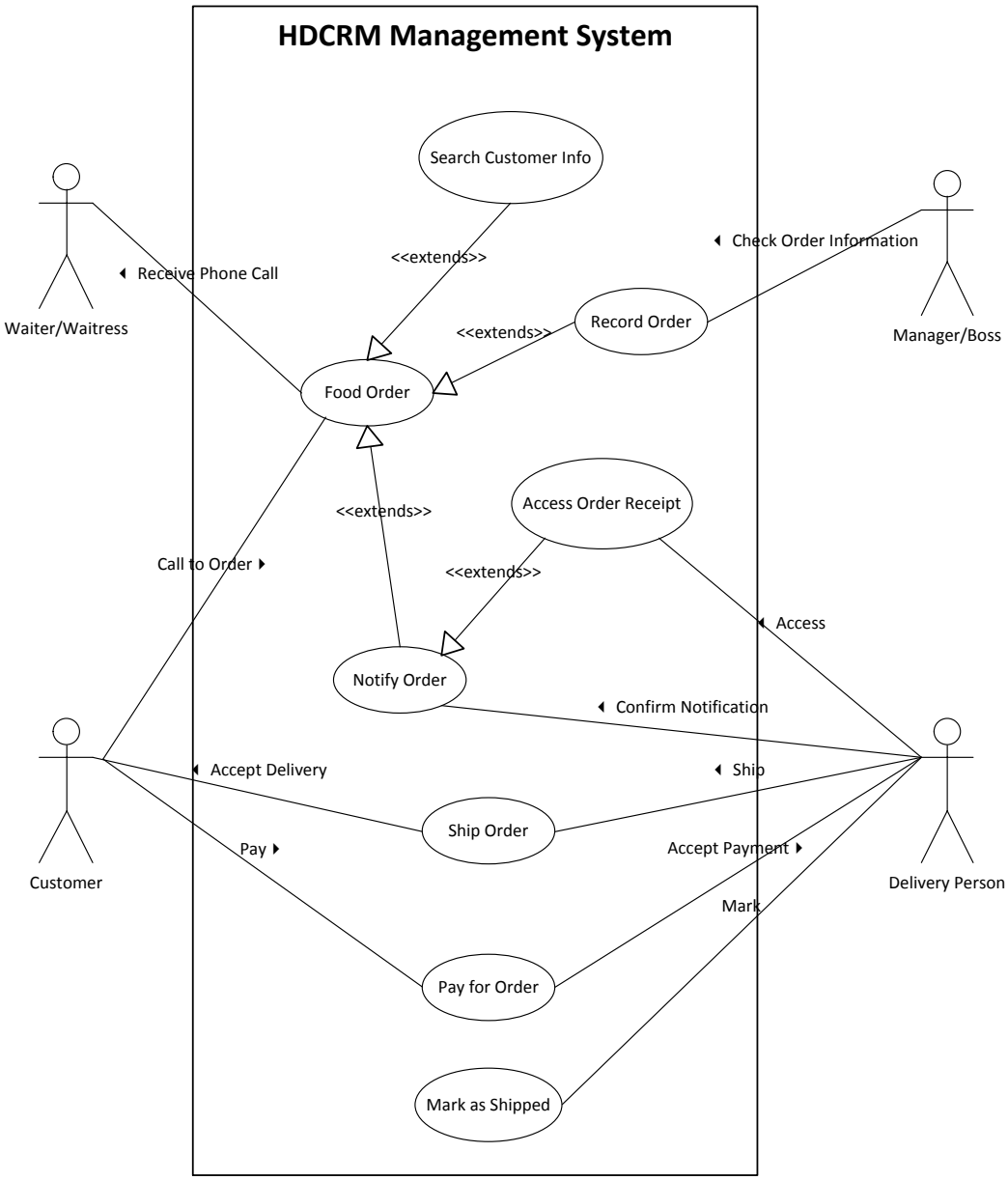

FIGURE 3. Use Case Diagram for HDCRM.

The Use Case diagram above defines four actors actively involved into home delivery business workflow and about seven use cases that are presented as each pieces of functionalities that the system has to take into account.

5.3.3 Sequence Diagram

The Sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Sequence diagram is considered as an extension of the Use Case diagram illustrated in the previous section. Thus, next figure depicts a sequence diagram for the detailed logic of interactions between objects within the HDCRM system.

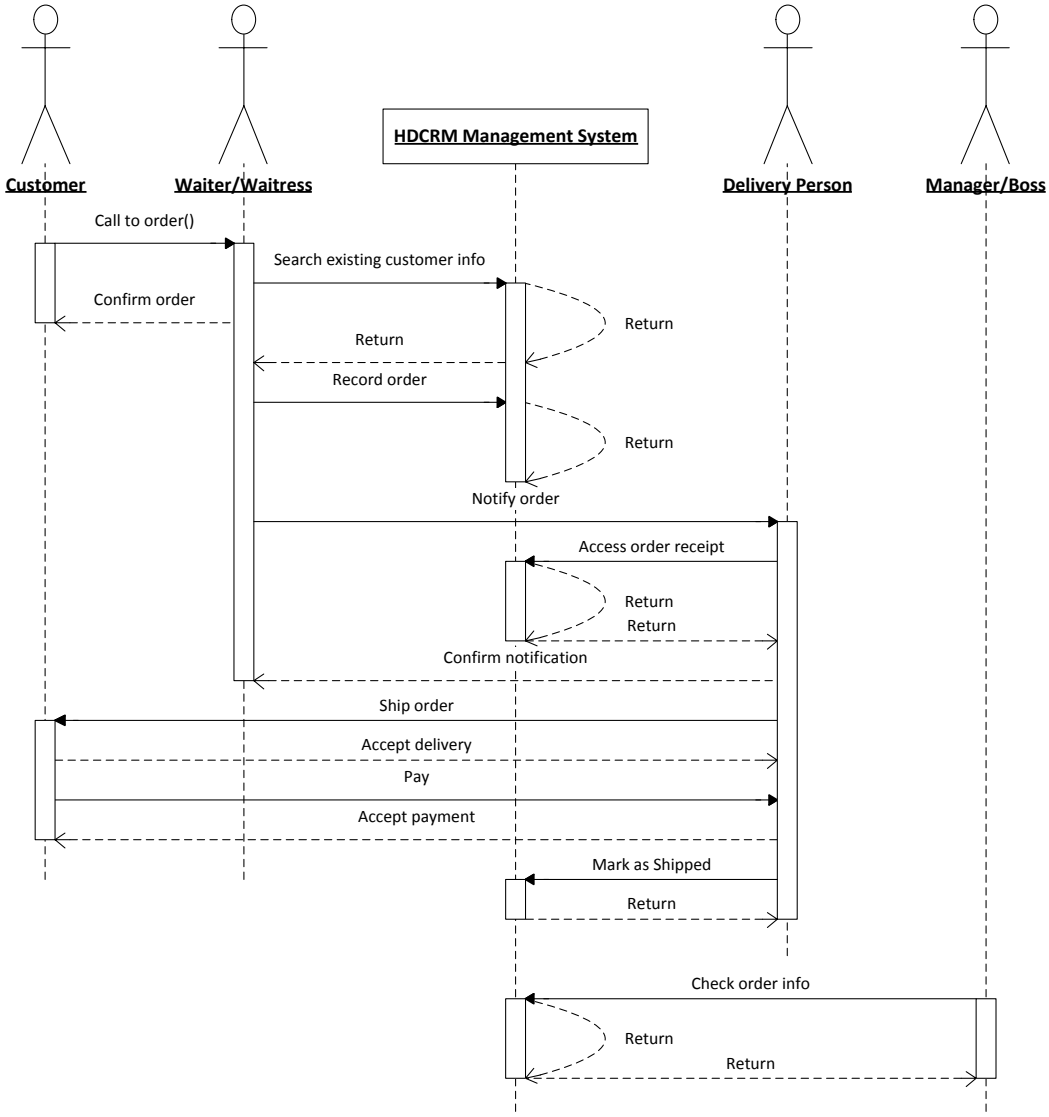

FIGURE 4. Sequence Diagram for HDCRM.

Figure 4 in the previous page illustrates a clear view of the flow of logic within the system. In this diagram, vertically those object life lines represent different processes or objects that live simultaneously, and horizontally the messages arrow lines defines what message exchanges between them and in which order they occur.

5.4 Systems Design

Systems design is always somewhat overlapped with Systems analysis while deeply investigating all system requirements. Much of these diggings have already been done through the investigation and analysis phases (Avison & Fitzgerald, 2006). However, in this paper, Systems investigation merely lists several requirements to be met literally, and in Systems analysis Section only object oriented UML modeling has conducted. In the following text, from two different aspects, data modeling and process modeling will be introduced for Systems design.

5.4.1 Entity-relationship (ER) Diagram

Data modeling emphasizes understanding and capturing real world concern in the Systems design. It is believed ER diagram is an important tool, among others, to achieve it (Avison & Fitzgerald, 2006). Teorey (1999) states in his book that ER model was a successful tool for communication between designer and end user during systems analysis and logical design, and it concentrates on representing semantic details in data relationships. The ER diagram revealed in next page tries to show most relevant entities and relationships between each participant in home delivery business operation in a small restaurant.

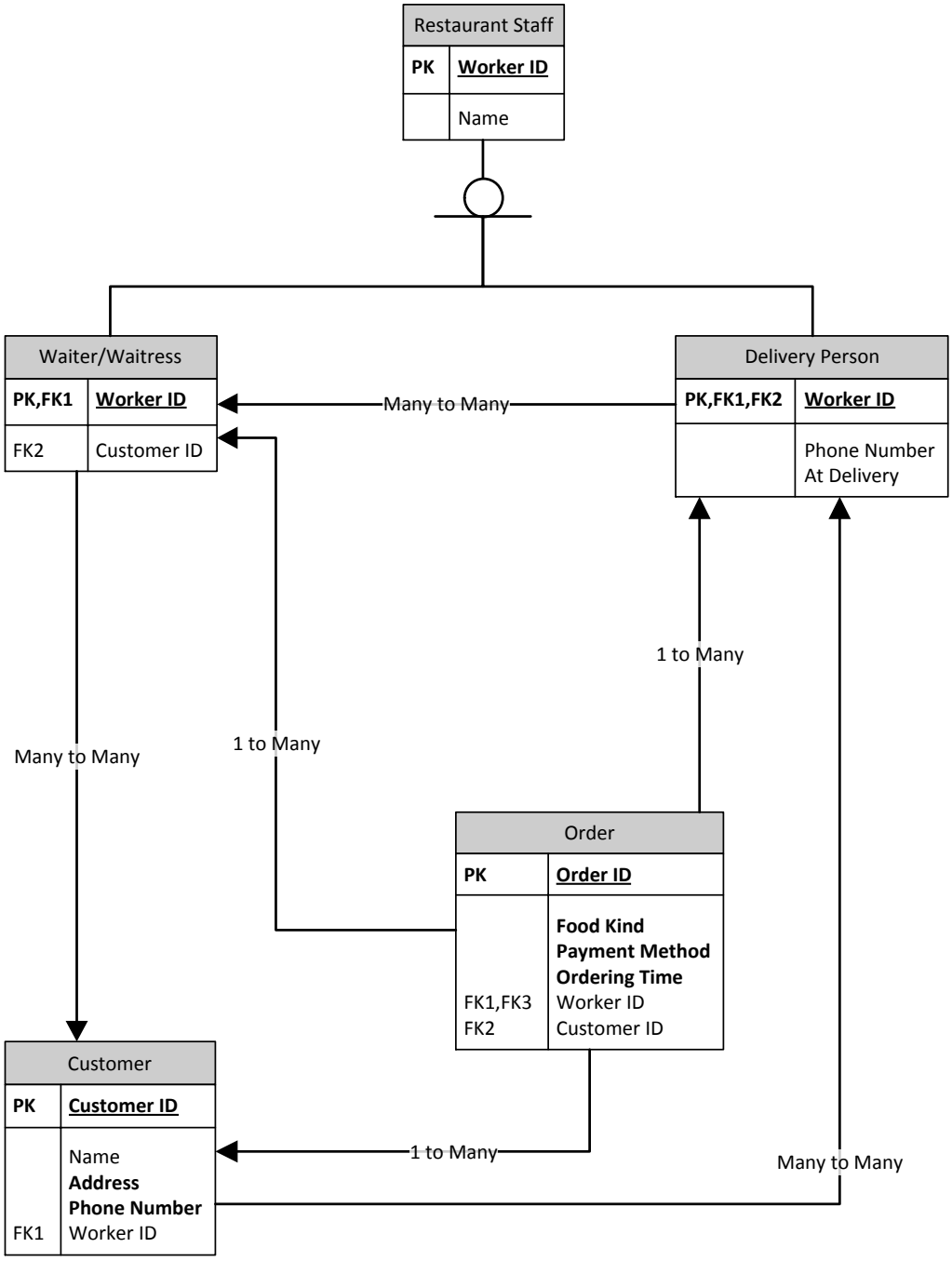

FIGURE 5. Entity-relationship data model diagram for HDCRM.

5.4.2 Data Flow Diagram (DFD)

DFD is regarded as a graphical representation of the data flow from the system logic aspect in an information system (Wikipedia, 2012). By specifying logical sequences of the system, DFD "*describes what a system will do rather than how it will be done*" (Avison & Fitzgerald, 2006). By presenting the following diagrams,

the author tries to specify what data flows in home delivery business operation in a small restaurant.

There are usually four steps in creating relatively rich DFD diagrams:

5.4.2.1 Step 1: Identify a series of activities that data flow occurs

According to the Use Case diagram and Sequence diagram presented in the previous sections, these activities can be identified as:

1. Customer call to order;
2. Waiter/Waitress Search Existing customer info;
3. W confirm order;
4. W record new order;
5. W notify order to Delivery Person;
6. D access order receipt;
7. D confirm notification;
8. D ship order;
9. C accept delivery;
10. C pay order;
11. D accept payment;
12. D mark order as Shipped;

Alternatively, there would be also:

13. Manager/Boss check overall orders information;

For better usage in the following text, these activities can then be grouped as:

1. Recording order: reprensents activities 1-4;
2. Notifying order: represents activities 5-7;
3. Shipping order: represents activities 8-12, and alternatively;
4. Checking order: represents activities 13;

After activities where data flow occurs are determined, next step is to generate a Context Level DFD diagram.

5.4.2.2 Step 2: Generate Context Level DFD diagram

Context Level DFD diagram represents very high level of DFD diagram. It presents both entities such as Customer, Waiter/Waitress, Delivery Person and Manager/Boss and all data that these external entities proceed flow into and out from the HDCRM system as shown in Figure 6.

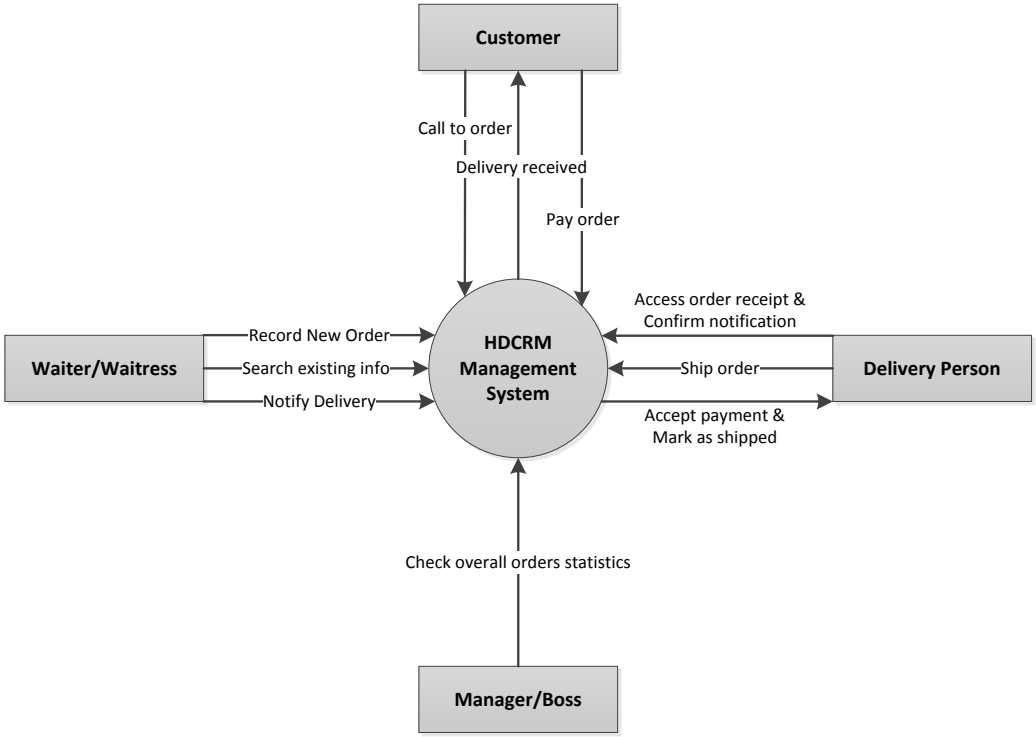

FIGURE 6. Context Level DFD diagram for HDCRM.

5.4.2.3 Step 3: Create Level 0 DFD diagram

Level 0 diagram is therefore a diagram showing all manageable sub processes within the HDCRM management system.

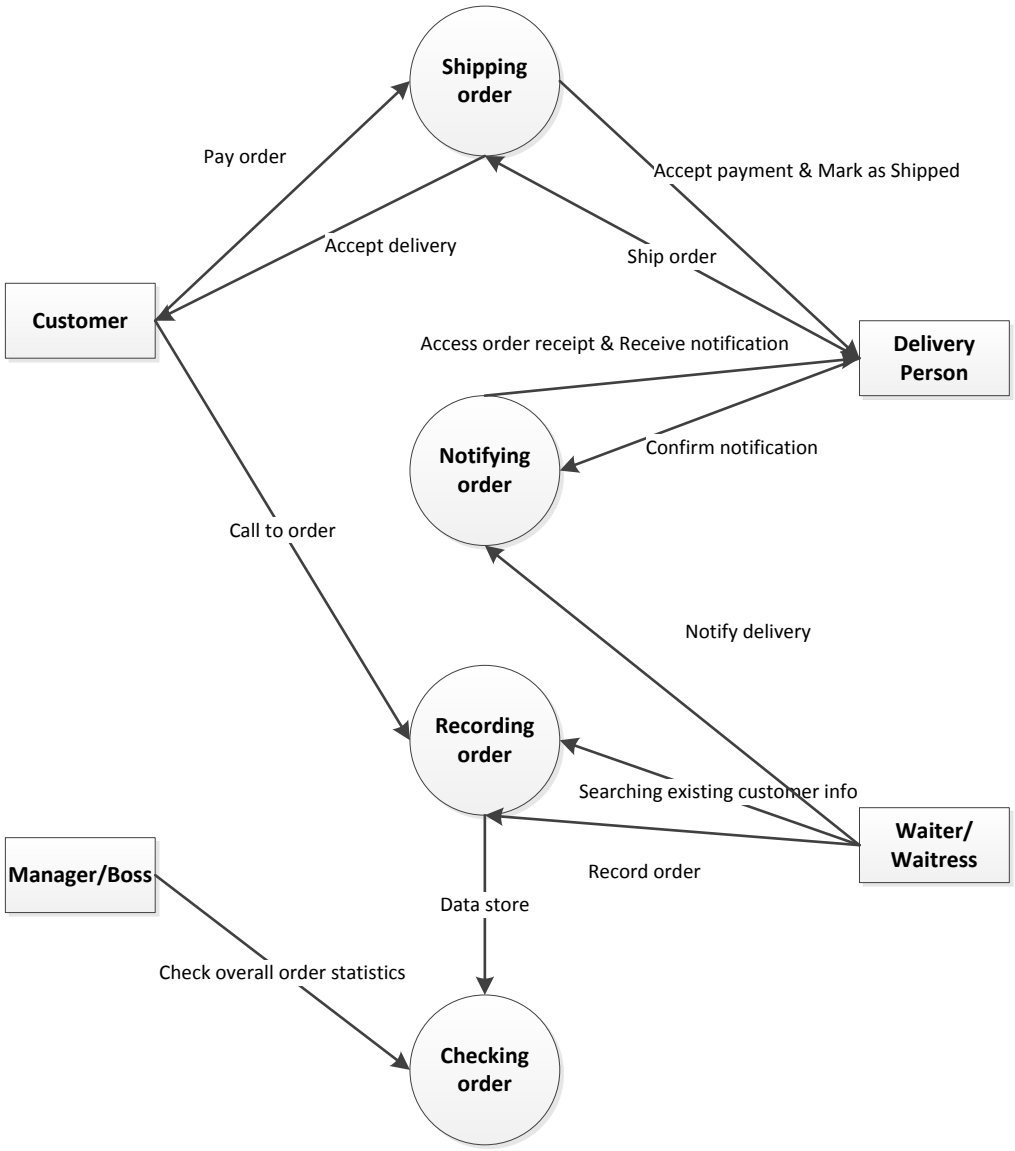

FIGURE 7. Level 0 DFD diagram for HDCRM.

5.4.2.4 Step 4: Construct Level 1 (Low Level) DFD diagram

Finally, by going through all steps above, the Low Level DFD diagram that determined actual data flows and data stores in home delivery business operation is presented in Figure 8 in next page.

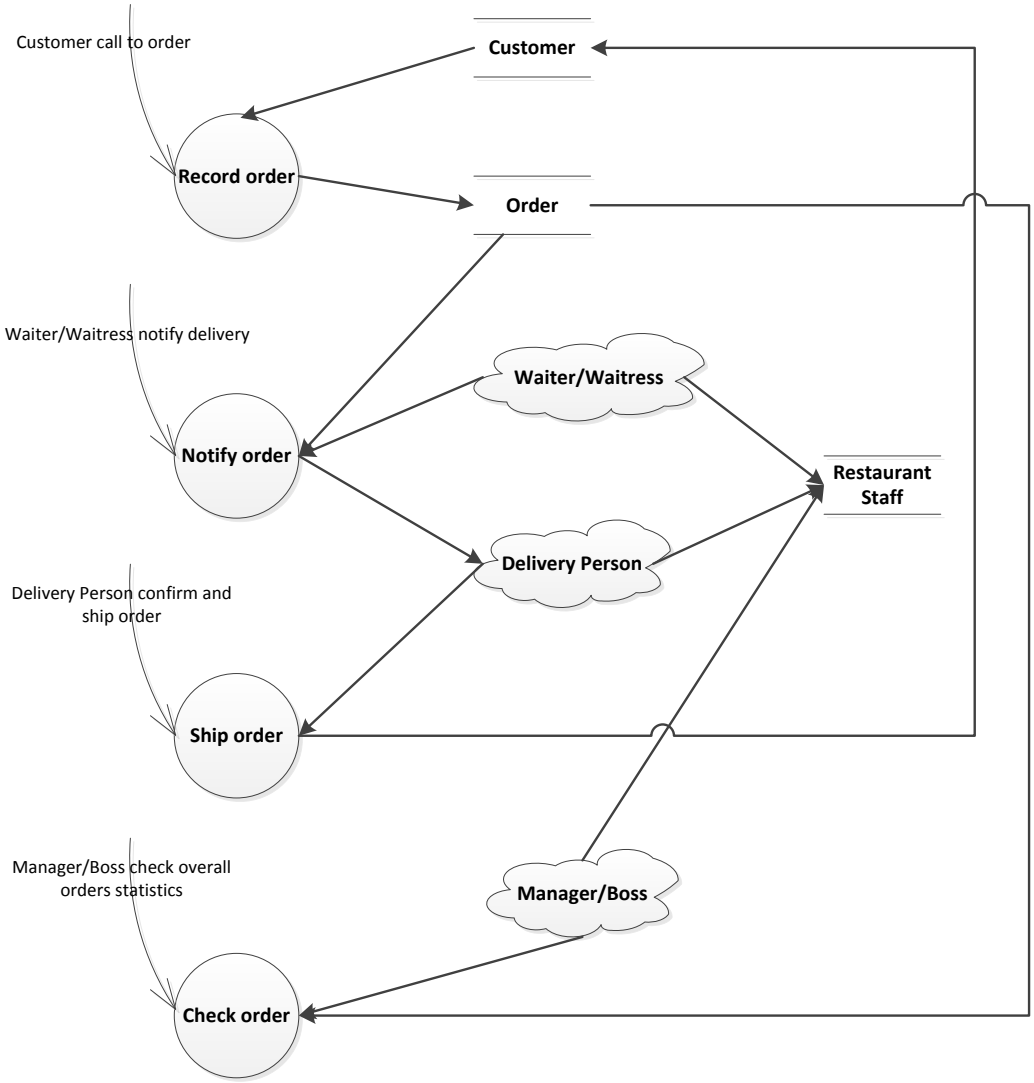

FIGURE 8. Level 1 (Low Level) DFD diagram for HDCRM.

5.5 Implementation

After all systems investigation, analysis and design have done, the following text describes implementation of such a system. As mentioned before, Drupal CMS is used as the Content Management Framework for the HDCRM system

implementation. However, this paper is not a how-to tutorial, thus the author is not going to introduce how to program with Drupal gradually. Instead, the following text tries to provide two most crucial modules that has developed and added to Drupal to fulfill the system requirements in order to meet business needs, though, there can be millions of possible ones to be made.

5.5.1 Module for Order Information

Drupal modules are plugins for Drupal that extend, build or enhance Drupal core functionality (Butcher, et al., 2010). As illustrated in the previous sections, entity "Order" plays as one of the most important roles in home delivery business operation. Thus, the first module provided here is a module that offers several functionalities including:

5.5.1.1 Administer order information

For Administors such as Manager/Boss in a restaurant, they might need full control of the HDCRM Management system. This functionality would give them possibility to manage fields of order information they would like to take by ticking link as presented in FIGURE 9 in next page.

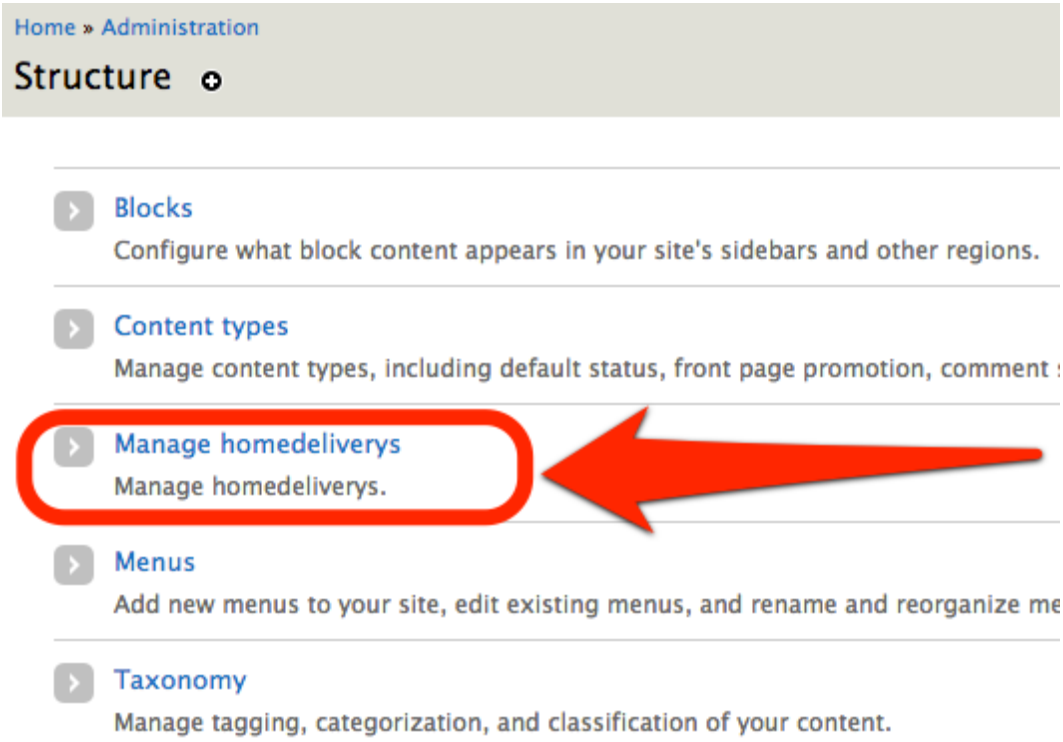

FIGURE 9. Access Point for administering order information.

Based on Entity-relationship diagram presented in the previous section, attributes such as customer name, address, phone number, food kinds and payment method in Class Order are demonstrated as shown in FIGURE 10:

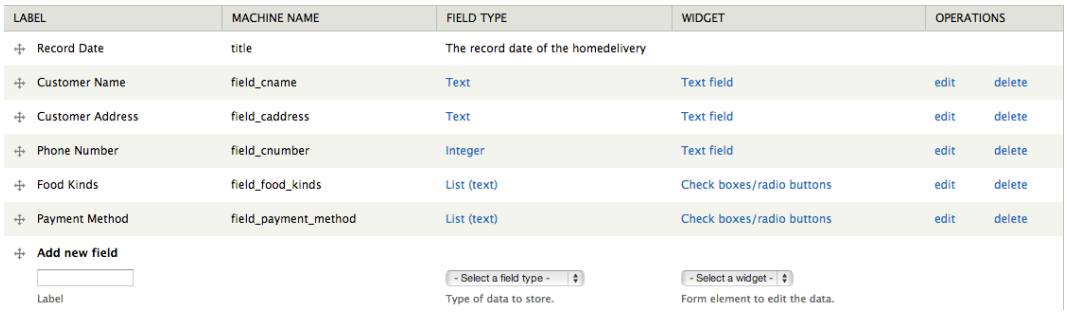

FIGURE 10. Administer order attributes.

To be frank, they are some shortages in current version of the HDCRM system. For instance, in order to validate inputs, customer phone number can be given as a customized field type rather than Integer. Nevertheless, the author believes that is just a matter of application development as the development process moves further.

Overall, this functionality would define possibility to administer order information for manager/boss in home delivery business operation. Moreover, other participants involved in home delivery business operation have not permission to do so.

5.5.1.2 Record order information

In home delivery business operation, Waiter/Waitress is regarded as the only party who creates an order in general. Therefore, such functionality would be assigned only to this role and even Manager/Boss would not be able to record an order to make system work constantly. Likewise, link to order information creation form should be presented to Waiter/Waitress at the first place, front page in the HDCRM Management system, as the following figure gives:

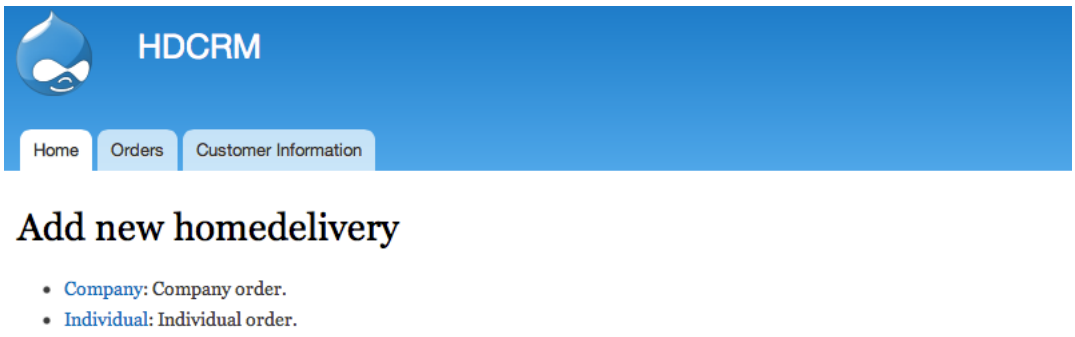

FIGURE 11. Access Point for recording order information.

By clicking “Individual” link in this front page, a form for gathering all order attributes of an order made by an individual person is presented as shown in the following figure:

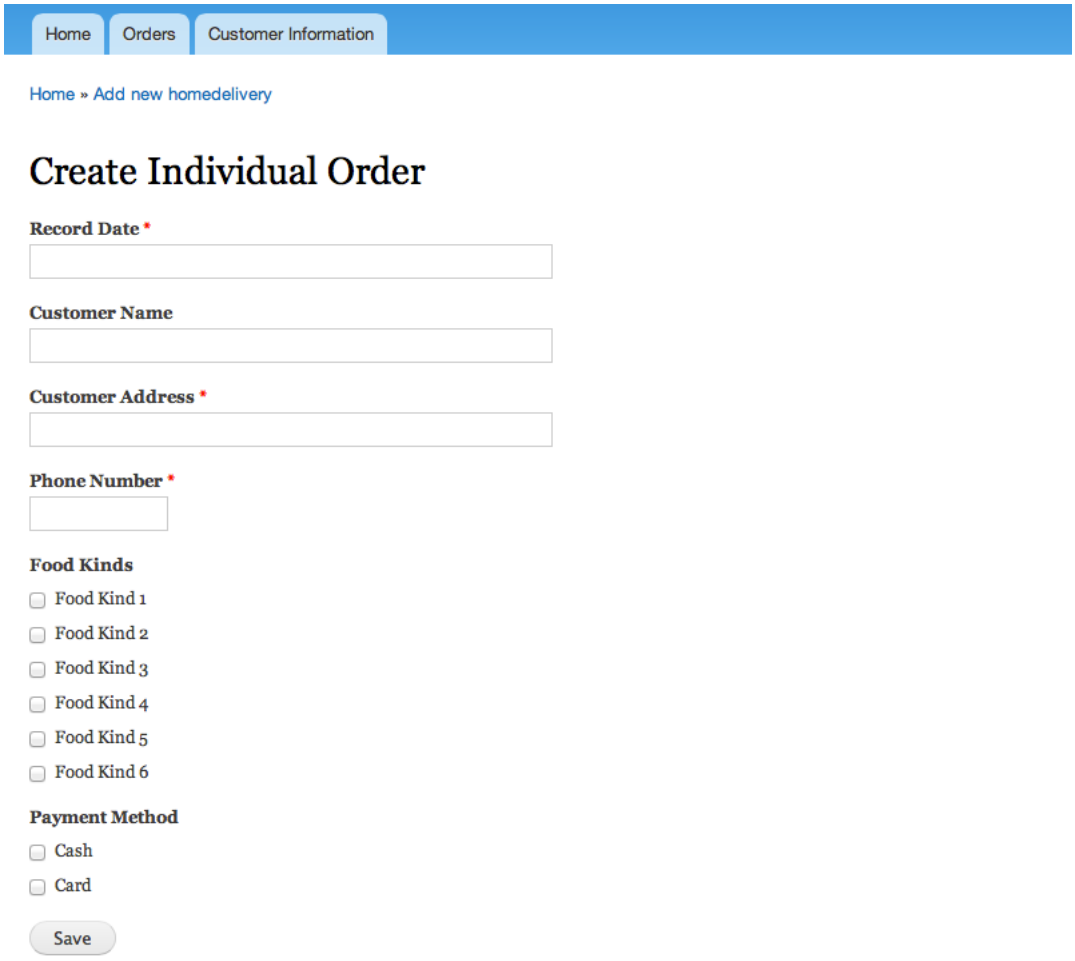

FIGURE 12. Form for recording order information.

After filling up and submitting this form, most valuable information has been added into database for any imaginable late usages.

5.5.1.3 View order information

View order information is one of the most frequent late usages of information records in database. Each participant in home delivery business operation would want to view order information for his/ her own purposes. Waiter/Waitress view order information and search through to filter existing customer information when he/she received phone calls from Customer. Similarly, Delivery Person would need to access order information when he/she is notified with new orders. Manager/Boss then would want to check overall order information daily, weekly, or monthly to analysis them for better decision-making and marketing in the future. In the HDCRM system demonstration in this paper, when no order has been created beforehand, this form will be redirected to front page as shown in FIGURE 10. However, when there are already orders in database, a tabular view of all orders made formerly will be presented as demonstrated in the following figure:

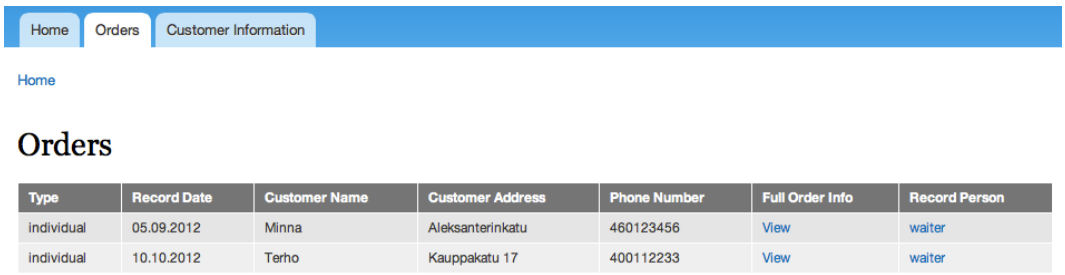

FIGURE 13. Tabular View of Order Information

Notice that in this tabular view of orders, most valuable order attributes are listed column by column. Besides, link to single order full-page view and link to person who recorded order information are presented.

5.5.1.4 Update order information

Afterwards, any order made should have the possibility to be modified and updated so as to correct mistakes. Regards to scope of current version of the HDCRM system, Waiter/Waitress is the only person who can modify or update order information. There might be also situation that Delivery Person finds out mistakes before Waiter/Waitress do. However, permission should be limited to Waiter/Waitress only in order to make the system working consistently. Thus, Waiter/Waitress can correct mistakes himself/herself, or he/she has to be informed of mistakes that need to be corrected. For instance, in FIGURE 13, only street name of customer Minna's address has been recorded, but street number is left out. Waiter/Waitress can modify and update such changes by going to "Edit" form as shown below.

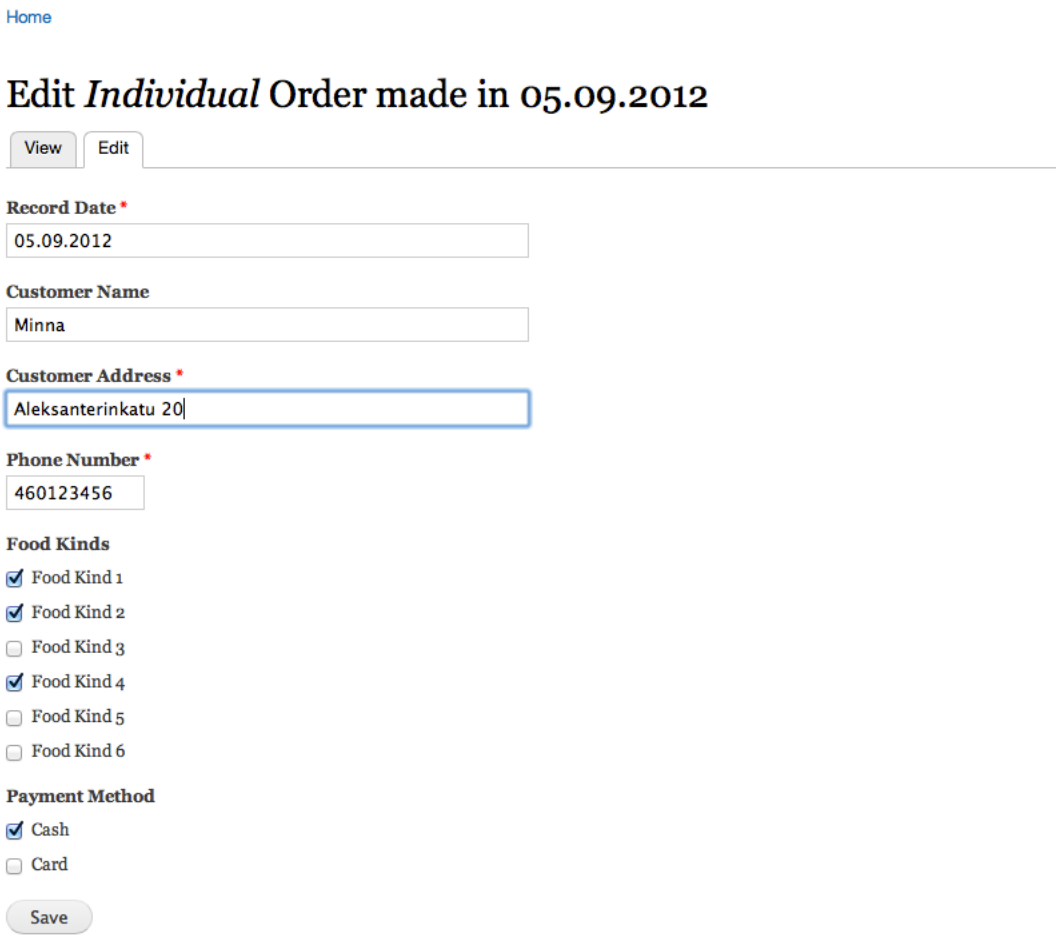

FIGURE 14. Edit form of Order Information.

After Waiter/Waitress correct mistakes like this and save it, information in database is therefore updated for further maintenance.

5.5.1.5 Delete order information

It is not recommended to delete order information in case someone executes this functionality accidentally resulting in a loss of important data. However, it can be a good usability design to keep such kind of functionality. Moreover, it is possible to make some changes in code so as to let system notify Manager/Boss, the only role that has permission to do so, with a confirmation form to avoid unintentional deleting action. FIGURE 15 presents this confirmation form when 'Delete' button was clicked.

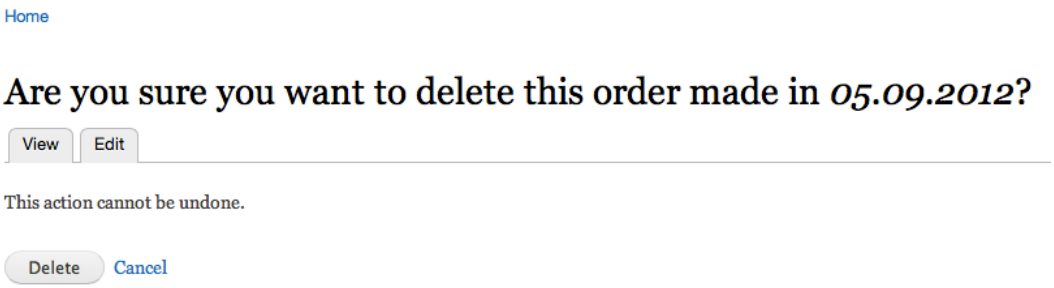

FIGURE 15. Confirmation form for deleting order information action.

Giving users such a confirmation form would let them not worry about deleting important information negligently.

5.5.2 Module for Customer Information

This module is designed for Waiter/Waitress when they receive phone calls from customers. As mentioned in the previous sections, phone number is unique to

identify which customer made which order. For this reason, this module is programmed by using phone number as searching criteria to search existing customer information.

At the very first point, only a form, with one textfield and one submit button, is presented to Waiter/Waitress, as shown in the following figure:

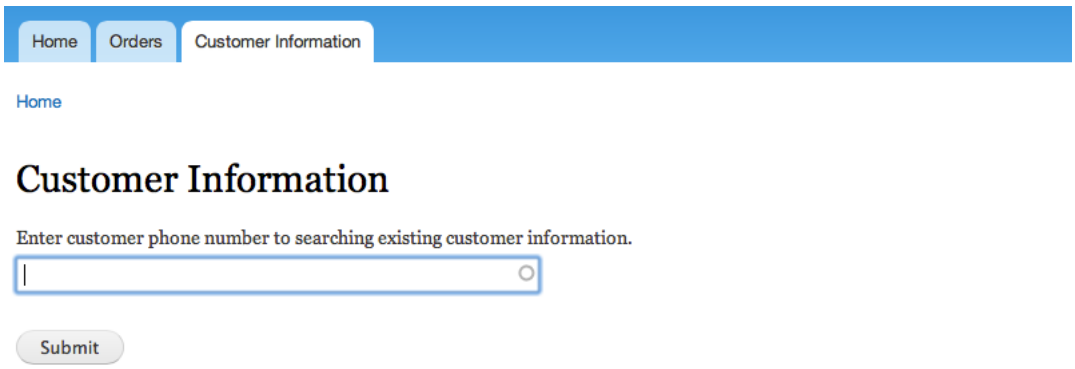

FIGURE 16. Form for searching existing customer information.

However, the magic remains already under cover in the textfield to be inputted. For example, there were already two order instances that have been recorded as presented previously in FIGURE 13. Phone numbers in these two order instances are 460123456 and 400112233. When Waiter/Waitress input number 4, a Drupal function called ”drupal_json_output” will give out jQuery-like result as shown in the following figure.

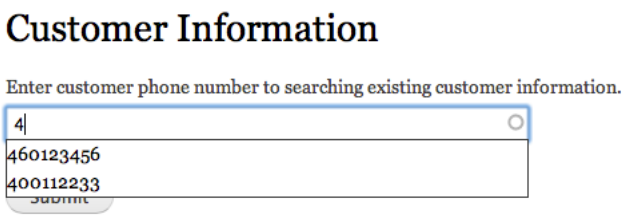

FIGURE 17. JSON output of textfield with two phone numbers.

As FIGURE 17 illustrates, both of the two phone numbers is given out. Nevertheless, once Waiter/Waitress input one more number “6” into the textfield, only one phone number will be given out as shown in the figure below:

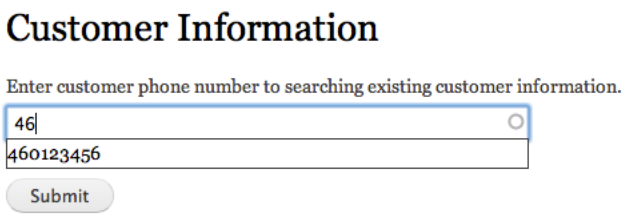

FIGURE 18. JSON output of textfield with only one phone number.

As this module is not yet fully developed, the submitted form will merely lead to a result of one line of text as shown in the next figure.

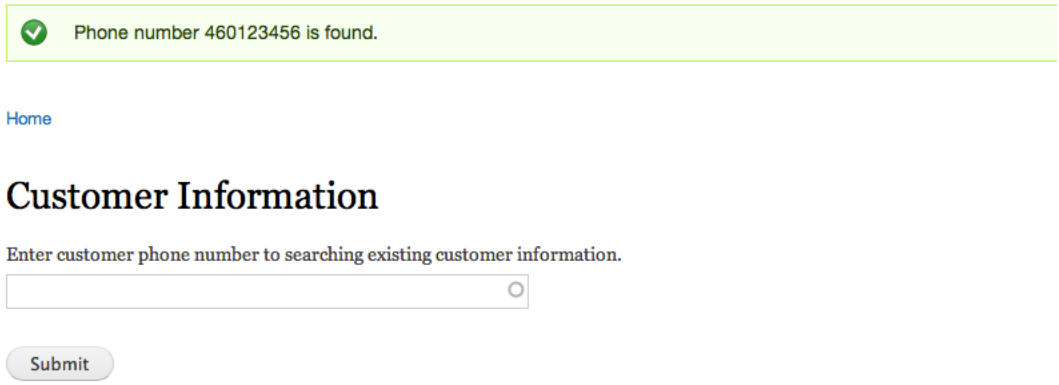

FIGURE 19. Output result of submission.

However, to make this HDCRM system a constant application, this module should follow a design flowchart as the figure illustrates as follows:

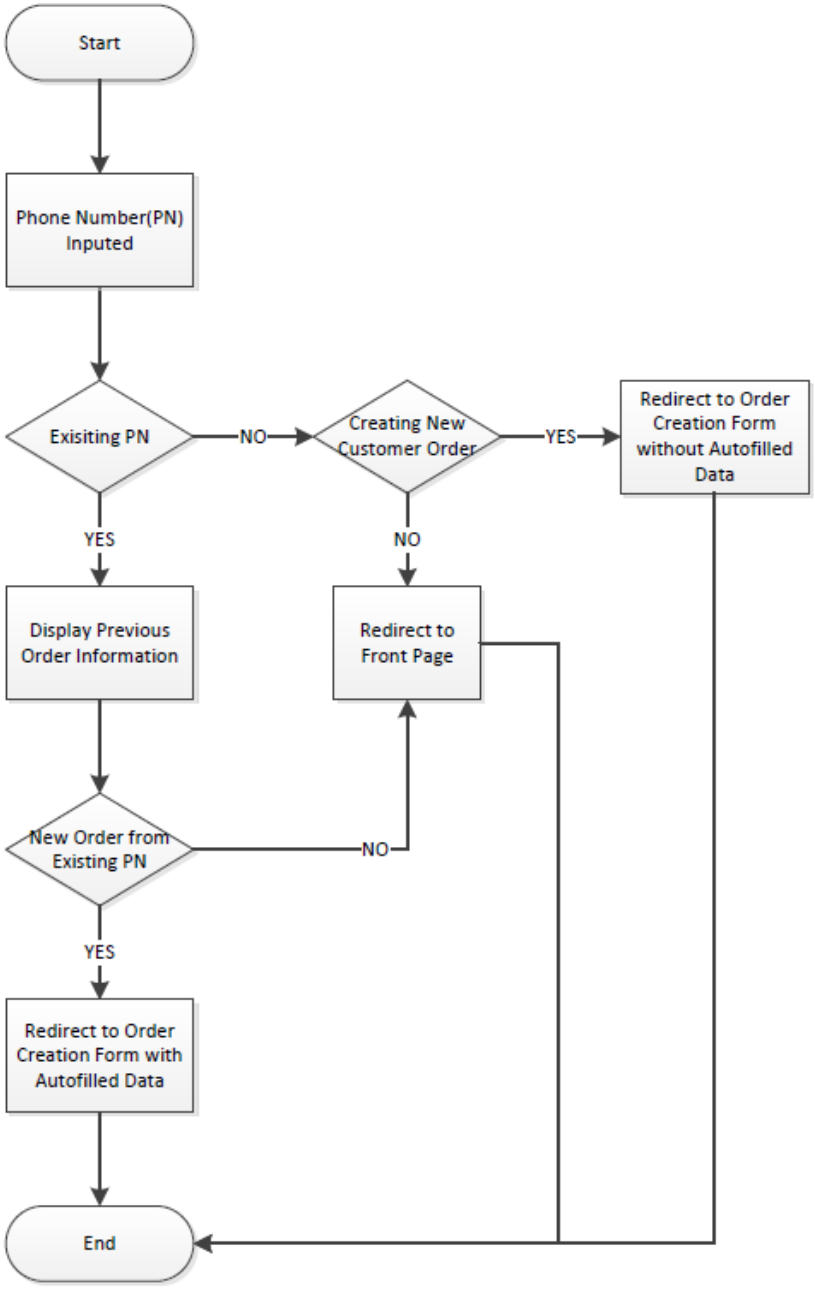

FIGURE 20. Design Flowchart for Customer Information Module.

Nevertheless, the output of submission shown in FIGURE 19 makes sense to prove that fulfilling needs predefined in the design flowchart above is possible and is only a matter of time.

5.5.3 Other Modules

There are much more business needs in home delivery business operation other than the two modules presented previously. Some possible modules to be developed in the future are assumed as follows according to different business needs:

1. A module that handling statistics of overall order information so as to let Manager/Boss have an easy and comfortable view of these statistics;
2. A module for Delivery Person to access order information, especially a map view of routine from starting point to destination. Moreover, it would give functionality for Delivery Person to mark order as shipped so as to remind Waiter/Waitress about shipped deliveries, Furthermore, it can provide data such as starting time and ending time of deliveries in order to let system perform statistics analysis.
3. A module for displaying order information, especially a list of food kinds, to screen for kitchen to produce foods. Although in the research scope of this paper, kitchen is not included, it is believed that a rich enough management system need to take this participant in restaurant business into account.

All in all, to fulfill business needs in home delivery business operation by providing Drupal modules with certain functionalities, a good design flow must be followed.

5.6 Review and Maintenance

Since this paper intends only to provide a prototype for home delivery business operation, and tries to find out what kind of application would fit such business needs in a small restaurant, review and maintenance part will be left out of the research scope.

However, review and maintenance of application is truly a very important part of software application development. Enough time is deserved to be spent in this stage of development to complete all processes in development cycle. Additionally, all these development processes should be iterated repeatedly to make this application better functionality.

6 DATA ANALYSIS

As research question in Introduction section of this paper declares, this paper tries to find out what kind of SaaS application for CRM would work in home delivery business operation in a small restaurant. In the previous three chapters, problems in home delivery business operation were already pointed out. Meanwhile, the HDCRM artifact prototype with several presets including what business needs there are in home delivery business operation is presented. However, no data illustrating workflows and time spent in each activity in home delivery business operation in real life has been given yet. Thus, that is what this chapter is going to be about.

6.1 Workflows and Time Spent

The author spent three days (one weekday and two weekend days) in a restaurant (Restaurant A in text below) and recorded all activities related to home delivery business operation in Restaurant A between 12.00 at noon and 20.00 at night.

There are in general the following several activities in a home delivery workflow:

1. Waiter:
    a. Receive phone call & Record order on paper (1a in text below)
    b. Notify kitchen food kinds in this order (1b)
    c. Notify Delivery Person via phone (1c)
2. Kitchen:
    a. Food ready to go (2a)
3. Delivery Person:
    a. Shipping start time (3a)
    b. Shipping end time (3b)
    c. Back to restaurant time (3c): N/A indicates no return between two orders delivery time

The graphs in following pages illustrate these activities occurred in Restaurant A in these three days.

6.1.1 Day 1

This section presents a table view of time spent for each activity for each order in Day 1 (weekday). Some of these orders come in quite a close time, thus Delivery Person would have to handle two or three orders in one same queue and not come back to restaurant during that.

TABLE 1. Time spent for each activity in Day 1.

	1a	1b	1c	2a	3a	3b	3c
Order 1	12.00	12.05	12.06	12.22	12.22	12.35	12.48
Order 2	15.00	15.05	15.06	15.16	15.16	15.36	**15.56**
Order 3	**15.35**	15.37	**15.38**	15.47	16.00	16.14	N/A
Order 4	15.40	15.45	15.46	15.55	16.14	16.28	N/A
Order 5	15.40	15.45	15.47	16.00	16.28	16.41	17.00
Order 6	18.00	18.05	18.06	18.21	18.27	18.35	N/A
Order 7	18.15	18.20	18.21	18.27	18.35	18.42	**18.51**
Order 8	**18.35**	18.38	**18.39**	18.41	18.51	19.01	19.11
Order 9	19.05	19.10	19.11	19.18	19.18	19.25	19.32
Order 10	19.35	19.40	19.41	20.00	20.16	20.24	N/A
Order 11	19.50	19.53	19.54	19.58	20.25	20.30	N/A
Order 12	20.00	20.05	20.06	20.16	20.31	20.41	20.55

Notice that when customers of order 3 and order 8 call for home deliveries, Delivery Person is still on his/her way shipping previous ones. In this kind of situations, it is difficult for both Waiter/Waitress and Delivery Person to communication with each other. Only phone call would make sense for either of them to get to know what is going on. Nevertheless, phone calls to solve these issues are costly and are time waste for Waiter/Waitress especially when it is busy for them to serve customers having meals inside restaurant base.

6.1.2 Day 2

There are relatively few orders in Day 2 (first weekend day). However, it can be noticed that there still is same problem as illustrated in the previous section. While Delivery Person is still on his/her way of shipping, new order comes in at that same time.

TABLE 2. Time spent for each activity in Day 2.

	1a	1b	1c	2a	3a	3b	3c
Order 1	12.55	13.00	13.01	13.10	13.22	13.35	N/A
Order 2	13.14	13.17	13.18	13.22	13.36	13.54	14.02
Order 3	14.15	14.20	14.21	14.30	14.40	15.04	N/A
Order 4	14.30	14.32	14.33	14.40	15.05	15.20	**15.25**
Order 5	**14.53**	14.55	**14.56**	15.10	15.25	15.35	15.45
Order 6	16.45	16.50	16.51	18.35	18.35	19.00	19.15
Order 7	19.12	19.18	19.19	19.25	19.25	19.32	19.39
Order 8	19.35	19.40	19.41	19.50	19.50	20.00	N/A
Order 9	19.54	19.57	19.58	19.50	20.00	20.10	20.20

6.1.3 Day 3

Table 3 in the next page presents as well time spent for each activity for each order in a weekend day. Compare to Day 2, there were much more orders in this weekend day. Therefore, problem of communication between Waiter/Waitress and Delivery Person became more observable. As shown in Table 3 in next page, Order 5, 6, 7 and 8 were made almost at the same time, every restaurant staff started to be so tense to handle his/her own work. This would result in that a management system for these participants in home delivery business operation is something necessary and can ease these workers' job.

TABLE 3. Time spent for each activity in Day 3.

	1a	1b	1c	2a	3a	3b	3c
Order 1	12.00	12.05	12.06	12.15	12.35	12.55	N/A
Order 2	12.02	12.07	12.08	12.35	12.55	13.10	13.25
Order 3	14.50	14.55	14.51	15.05	15.05	15.12	15.19
Order 4	15.25	15.30	15.31	15.40	15.40	16.00	16.20
Order 5	16.50	16.55	17.06	17.08	17.13	17.33	N/A
Order 6	16.55	17.00	17.06	17.10	17.33	17.48	N/A
Order 7	17.00	17.05	17.06	17.13	17.48	17.58	**18.02**
Order 8	**17.05**	17.10	**17.11**	17.30	18.02	18.12	N/A
Order 9	**17.45**	17.50	**17.51**	18.00	18.12	18.17	**18.25**
Order 10	**18.10**	18.15	**18.16**	18.25	18.40	18.45	N/A
Order 11	18.30	18.35	18.36	18.40	18.45	18.55	19.05
Order 12	19.55	20.00	20.01	20.10	20.10	20.25	20.40

In a word, these three tables illustrate workflows and time spent for each activity that occurs in home delivery business operation. Besides, they point out that information communication problems exist in this kind of restaurant business. Furthermore, they lead to a result of having a management system that can solve these problems and ease restaurant worker's job.

6.2 Benefits of HDCRM System

The purposeful artifact, HDCRM system, is designed for benefit growth and improvement of restaurant business in home delivery business operation. It is expected that the HDCRM system would have at least the following four positive impacts in home delivery business operation in a small restaurant.

6.2.1 Information Communication among Restaurant Workers

As illustrated with three tables showing workflows and time spent for each activity for each order in three different days, the previous section indicates that there are problems of information communications that exist in real life home delivery business operation in Restaurant A.

Hence, the HDCRM system presented in the previous chapter brings a solution to this kind of problems as this system includes functionality of notifying Delivery Person of order information as well as that of marking orders that are already shipped.

6.2.2 Information Management for Waiter/Watress and Manager/Boss

Similarly, the HDCRM system consists of several databases where information of both customers and previous orders are stored. This gives the Waiter/Waitress possibilities to search existing customer information. Additionally, it would give the Manager/Boss feasibilities to check overall order information and statistics of these orders and customer information.

6.2.3 Easy Plugins for System Enhancement

As mentioned above, in addition to the Module for Order Information and Module for Customer Information, there are numerous other possible modules to be developed according to restaurant business context. These plugins are easy to be developed to enhance system functionalities and performance once specification of business needs is well predefined.

6.2.4 Possible System Integration with External Systems

Integration with current systems can take place with many different external systems. Some of these external systems include the banking system, E-commercial system, and warehouse stock system. The following text will take the E-commercial system as an example.

Phone calls are the unique means for customers to order home deliveries in Restaurant A right at this moment. This type of communication is, and in a long period will still be, such a stable means to connect customers with Restaurant A, and keep home delivery business operation continuous.

However, an E-commercial web system could be taken into consideration so as to provide one other approach for customers to make home delivery orders and then customer range can be expanded. Thereafter, system integration between these two solo systems becomes an interesting task.

6.3 Deficiencies of HDCRM System

Although the author believes the HDCRM system would be such a great system to handle Customer Relationship Management in home delivery business operation, the system might still include some drawbacks that hinder migration from current manual paper system to computer-based system. These deficiencies might involve the following two, among others.

6.3.1 Certain Equipment Required

There is a concern that it is difficult for the Waiter/Waitress to use the HDCRM system while they have to keep phone in hand. Thus, one possible solution for this issue would be hands free equipment. With hands free equipment, the Waiter/Waitress can reach customer phone calls anytime and anywhere within a certain distance to the host phone machine. Furthermore, with both hands free, the Waiter/Waitress can then focus on recording order information into the HDCRM system.

6.3.2 Sufficient Internet Connection Required

To give a demonstration, the SaaS artifact described above was hosted in a local host rather than a remote host. That is, Internet connection has not been taken into consideration during the research and development process.

However, Internet connection is compulsory when using the HDCRM system in real life. As stated in the previous text, the Waiter/Waitress needs to notify the Delivery Person about order information via the HDCRM system and vice versa. Similar situations are all over the place. Sufficient Internet connection is required for transferring all this information. Internet connection thus becomes one of the most significant necessities for adoption of the HDCRM system in real life.

7 CONCLUSIONS

In this chapter, the author commits himself to summarizing three findings after the whole research process, discussing reliability and validity of this research, and finally declaring limitation of research scope of this paper and providing several further study recommendations.

7.1 Findings

7.1.1 Finding 1. SaaS is adoptable for SMBs

As the literature review of this thesis points out, many IT suppliers regard SaaS as a serious alternative of software. Featured as pay as go, SaaS has potentials to help SMBs increase their business growth. It is believed that SaaS in SMBs will increase dramatically in the near future. Characteristics of SaaS determine that it can especially solve problems existing in SMBs. The author believes that home delivery business operation in small restaurants is one such kind of SMB and it asks for SaaS for internal management work.

7.1.2 Finding 2. Home delivery business demands CRM System

Home delivery business operation in small restaurants is one small part of the restaurant business. However, Customer Relationship Management is especially needed to be taken care of in this small part of the restaurant business. Through deep research and investigation in the previous chapters, the author found out that customer information and order information are important in retaining current business relationships and creating new ones. Thereafter, the author concludes a CRM system can help small restaurants in this particular part of business operation.

7.1.3 Finding 3. HDCRM meets home delivery business needs

As stated in Finding 2, home delivery business operation in small restaurants demands a CRM system to help management work with customer and order information. By presenting real life workflow in home delivery business

operation, the author also points out several problems in home delivery business operation. What is more, through a relatively sophisticated HDCRM system development life cycle the author illustrates how the HDCRM system aims to meet business needs in home delivery business operation. The author believes the designed artifact, the HDCRM system, achieved its goal and met business needs in home delivery business operation.

7.2 Discussion

7.2.1 Reliability

In this thesis, the author continuously replicates research procedures by observing problems existing in home delivery business operation, using several design tools to investigate possibilities of implementing a HDCRM system and developing it with predefined settings of home delivery business context. Thereafter, problems addressed were illustrated again with real life data of home delivery business workflows and time spent. Since the experimental study and the designed artifact firmly yield the same result, the author thus concludes the designed artifact, the HDCRM system, solves problems addressed.

7.2.2 Validity

It is believed that research can be invalid even though it is reliable. Despite the fact that this research gives one HDCRM system prototype for Restaurant A according to its business context, the research results achieved is considered to be valid in all those small restaurants since they would have similar problems and business needs when operating home delivery businesses.

7.3 Limitation and Further Study

Although this paper produces three findings that demonstrate relations between SaaS and SMBs, relations between home delivery business needs and the HDCRM Management system, the author believes there are several significant limitations of this research.

First, since this research concentrates on providing the HDCRM artifact as an instance that can fulfill home delivery business needs, this specific artifact would not be capable of solving problems within other business contexts.

Second, due to resource costs, the HDCRM artifact demonstrated in this paper is only one prototype hosted in a local host rather than a remote host. Thus, this research limits on giving reliable solution of the HDCRM system to meet home delivery business needs.

At last, the HDCRM artifact presented was implemented with Drupal open source CMS, which means the designed artifact is used within web browsers, hence, limitations exist in implementing certain functionalities that would be compatible with mobile devices. The author therefore excluded implementation of these functionalities from the research scope.

With regard to the research framework applied in this paper and the research results achieved, the author would like to provide several recommendations for further studies.

First, SaaS is really a hot topic, and it might be the future of software industry and even the future of any industry. However, there rarely are studies that are about SaaS adoption in SMBs.

Second, in this paper, the author predefines settings for the HDCRM system and implements this artifact according to these settings. Nonetheless, it is believed that the end users' attitudes towards this kind of system are of vital importance. Thus, the author would recommend further research on this issue.

Furthermore, it is an interesting topic to research how much IT companies who would like to play a role in SaaS adoption are willing to perform SaaS application development in different business fields.

REFERENCES

Published References

Armbrust, M. ym., 2010. A View of Cloud Computing. *Communications of the ACM,* April, 53(4), pp. 50-58.

Avison, D. & Fitzgerald, G., 2006. *Information Systems Development: methodologies, techniques & tools.* 4th Edition toim. s.l.:McGraw-Hill Education.

Benlian, A., Hess, T. & Buxmann, P., 2009. Drivers of SaaS-Adoption - An Empirical Study of Different Application Types. *Business & Information System Engineering,* 1(5), pp. 357-369.

Blackburn, J. D. & Scudder, G. D., 1996. Time-based software development. *Integrated Manufacturing Systems,* 7(2), pp. 60-66.

Butcher, M. ym., 2010. *Drupal 7 Module Development.* Birmingham: Packt Publishing Ltd.

Buttle, F., 2009. *Customer Relationship Management.* Amsterdam: Elsevier Ltd..

Chee, B. J. S. & Franklin, C., 2010. *Cloud computing : technologies and strategies of the ubiquitous data center.* Boca Raton: CRC.

Duan, Y., 2012. *Value Modeling and Calculation for Everything as a Service (XaaS) Based on Reuse.* Milan, IEEE, pp. 162-167.

Faulkner, X., 2000. *Usability Engineering.* New York: PALGRAVE.

Gold, N., Knight, C., Mohan, A. & Munro, M., 2004. Understanding Service-Oriented Software. *Software, IEEE,* March, 21(2), pp. 71-77.

Hai, H. & Sakoda, S., 2009. SaaS and Integration Best Practices. *FUJITSU Scientific & Technical Journal,* July, 45(3), pp. 257-264.

Hevner, A. R., March, S. T., Park, J. & Ram, S., 2004. Design Science in Information Systems Research. *MIS Quarterly,* March, 28(1), pp. 75-105.

International Telecommunication Union (ITU), 2011. *ICT Facts Figures 2011,* Place des Nations: s.n.

Kepes, B., 2010. *Moving your Infrastructure to the Cloud,* s.l.: Diversity Limited.

Kotler, P., 2010. *Principles of marketing.* Upper Saddle River: Pearson.

Pring, B., 2005. *Adoption of Software as a Service Is Happening Outside of CRM,* s.l.: Gartner.

Software & Information Industry Association, 2001. *Software as a Service: Strategic Backgrounder,* Washington DC: s.n.

Software & Information Industry Association, 2011. *Guide to Cloud Computing for Policymakers,* s.l.: SIIA.

Taylor, M. J., England, D. & Gresty, D., 2001. Knowledge for Web site development. *Internet Research,* 11(5), pp. 451-461.

Teorey, T. J., 1999. *Database Modeling & Design.* 3rd Edition toim. San Francisco: Morgan Kaufmann Publishers, Inc.

Todd, S., 1999. A guide to the Internet and World Wide Web. *Structural Survey,* 17(1), pp. 36-41.

Tucker, A., Morelli, R. & Silva, C. D., 2011. *Software Development.* s.l.:CRC Press.

Winkler, V. J., 2011. *Securing the Cloud: Cloud Computer Security Techniques and Tactics.* Waltham: Elsevier Inc.

Electronic References

Box, J., 2012. *Software as a Service: an old name for rehashed offerings?*. [Online]
Available at:
http://www.cloudpro.co.uk/julian-box/2959/software-service-old-name-rehashed-offerings
[Haettu 5 October 2012].

MaaS360, 2011. *Mobility and SaaS: Two great trends that work great together.* [Online]
Available at:
http://www.maas360.com/maasters/blog/mobilitymanagement/mobility-and-saas-two-great-trends-that-work-great-together/
[Haettu 5 October 2012].

Pettey, C., 2011. *Gartner Reveals Top Predictions for IT Organizations and Users for 2012 and Beyond.* [Online]
Available at: http://www.gartner.com/it/page.jsp?id=1862714
[Haettu 5 October 2012].

Salesforce.com, 2008. *The 7 Secrets of SaaS Startup Success.* [Online]
Available at: www.salesforce.com/assets/pdf/misc/WP_7Secrets_0408.pdf
[Haettu 15 Augest 2012].

Salesforce, 2012. *CRM from Salesforce.com - No Hardware, No Software, No Headache - salesforce.com.* [Online]
Available at: http://www.salesforce.com/crm/
[Haettu 24 October 2012].

Smith, D. M. ym., 2011. *Predicts 2012: Cloud Computing Is Becoming a Reality.* [Online]
Available at: http://www.gartner.com/id=1870118
[Haettu 5 October 2012].

Wikipedia, 2012. *Cloud computing.* [Online]
Available at: http://en.wikipedia.org/wiki/Cloud_computing
[Haettu 5 October 2012].

Wikipedia, 2012. *Data flow diagram.* [Online]
Available at: http://en.wikipedia.org/wiki/Data_flow_diagram
[Haettu 11 October 2012].

Wikipedia, 2012. *Dot-com bubble.* [Online]
Available at: http://en.wikipedia.org/wiki/Dot-com_bubble#cite_note-1
[Haettu 4 October 2012].

Wikipedia, 2012. *Salesforce.com - Wikipedia, the free encyclopedia.* [Online]
Available at: http://en.wikipedia.org/wiki/Salesforce
[Haettu 25 October 2012].

Wikipedia, 2012. *Software as a service.* [Online]
Available at: http://en.wikipedia.org/wiki/Software_as_a_service
[Haettu 5 October 2012].

APPENDICES

APPENDIX 1: Codes of Module for Order Information

homedelivery.info

```
name = Home Delivery Record Information
description = Record home delivery information from customer phone call.
core = 7.x
package = HDCRM
files[] = homedelivery.module
files[] = homedelivery.pages.inc
files[] = homedelivery.admin.inc
files[] = homedelivery.install
files[] = homedelivery.controller.inc
dependencies[] = field_ui
```

homedelivery.install

```
<?php

function homedelivery_schema() {
  $schema['homedelivery'] = array(
    'description' => 'The base table for homedeliverys.',
    'fields' => array(
      'aid' => array(
        'description' => 'The primary identifier for an homedelivery.',
        'type' => 'serial',
        'unsigned' => TRUE,
        'not null' => TRUE,
      ),
      'vid' => array(
        'description' => 'The current {homedelivery_revision}.vid version
identifier.',
        'type' => 'int',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'default' => 0,
      ),
      'type' => array(
        'description' => 'The {homedelivery_type} of this homedelivery.',
        'type' => 'varchar',
        'length' => 32,
        'not null' => TRUE,
        'default' => '',
      ),
      'title' => array(
        'description' => 'The title of this homedelivery.',
        'type' => 'varchar',
        'length' => 255,
        'not null' => TRUE,
        'default' => '',
      ),
      'created' => array(
        'description' => 'The Unix timestamp when the homedelivery was created.',
        'type' => 'int',
        'not null' => TRUE,
        'default' => 0,
      ),
      'changed' => array(
        'description' => 'The Unix timestamp when the homedelivery was most recently
saved.',
        'type' => 'int',
        'not null' => TRUE,
        'default' => 0,
      ),
```

```
    ),
    'unique keys' => array(
      'aid_vid' => array('aid', 'vid'),
      'aid'     => array('aid')
      ),
    'primary key' => array('aid'),
  );

  $schema['homedelivery_revision'] = array(
    'description' => 'Stores information about each saved version of an
{homedelivery}.',
    'fields' => array(
      'aid' => array(
        'description' => 'The {homedelivery} this version belongs to.',
        'type' => 'int',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'default' => 0,
      ),
      'vid' => array(
        'description' => 'The primary identifier for this version.',
        'type' => 'serial',
        'unsigned' => TRUE,
        'not null' => TRUE,
      ),
      'title' => array(
        'description' => 'The title of this version.',
        'type' => 'varchar',
        'length' => 255,
        'not null' => TRUE,
        'default' => '',
      ),
      'created' => array(
        'description' => 'The Unix timestamp when the homedelivery was created.',
        'type' => 'int',
        'not null' => TRUE,
        'default' => 0,
      ),
    ),
    'indexes' => array(
      'aid' => array('aid'),
    ),
    'primary key' => array('vid'),
    'foreign keys' => array(
      'homedelivery' => array(
        'table' => 'homedelivery',
        'columns' => array(
          'aid' => 'aid',
        ),
      ),
    ),
  );

  return $schema;
}
```

homedelivery.module

```
<?php

/**
 * Implements hook_entity_info().
 */
function homedelivery_entity_info() {
  $return['homedelivery'] = array(
    'label' => t('Homedelivery'),
    'controller class' => 'HomedeliveryController',
    'base table' => 'homedelivery',
    'revision table' => 'homedelivery_revision',
    'uri callback' => 'homedelivery_uri',
    'fieldable' => TRUE,
```

```
    'entity keys' => array(
      'id' => 'aid',
      'revision' => 'vid',
      'bundle' => 'type',
      'label' => 'title',
    ),
    'bundle keys' => array(
      'bundle' => 'type',
    ),
    'static cache' => TRUE,
    'bundles' => array(),
    'view modes' => array(
      'full' => array(
        'label' => t('Full content'),
        'custom settings' => FALSE,
      ),
      'teaser' => array(
        'label' => t('Teaser'),
        'custom settings' => FALSE,
      ),
    ),
  );

  foreach (homedelivery_types() as $type => $info) {
    $return['homedelivery']['bundles'][$type] = array(
      'label' => $info->name,
      'admin' => array(
        'path' => 'admin/structure/homedeliverys/manage/%homedelivery_type',
        'real path' => 'admin/structure/homedeliverys/manage/' . str_replace('_',
'-', $type),
        'bundle argument' => 4,
        'access arguments' => array('administer homedeliverys'),
      ),
    );
  }

  return $return;
}

/**
 * Entity uri callback.
 */
function homedelivery_uri($homedelivery) {
  return array(
    'path' => 'homedelivery/' . $homedelivery->aid,
  );
}

/**
 * Implements hook_fied_extra_fields().
 */
function homedelivery_field_extra_fields() {
  $extra = array();

  foreach (homedelivery_types() as $type) {
    $extra['homedelivery'][$type->type] = array(
      'form' => array(
        'title' => array(
          'label' => t('Record Date'),
          'description' => t('The record date of the homedelivery'),
          'weight' => -5,
        ),
      ),
      'display' => array(
        'title' => array(
          'label' => t('Record Date'),
          'description' => t('The record date of the homedelivery'),
          'weight' => -5,
        ),
      ),
    );
  }

  return $extra;
}

/**
```

```
 * Implements hook_permission().
 */
function homedelivery_permission() {
  return array(
    'administer homedeliverys' =>  array(
      'title' => t('Administer homedeliverys'),
    ),
    'create homedeliverys' =>  array(
      'title' => t('Create homedeliverys'),
    ),
    'update homedeliverys' =>  array(
      'title' => t('Update homedeliverys'),
    ),
    'view homedeliverys' =>  array(
      'title' => t('View homedeliverys'),
    ),
    'delete homedeliverys' =>  array(
      'title' => t('Delete homedeliverys'),
    ),
  );
}

/**
 * Implements hook_menu().
 */
function homedelivery_menu() {
  $items['admin/structure/homedeliverys'] = array(
    'title' => 'Manage homedeliverys',
    'description' => 'Manage homedeliverys.',
    'page callback' => 'homedelivery overview types',
    'access arguments' => array('administer homedeliverys'),
    'file' => 'homedelivery.admin.inc',
  );
  $items['admin/structure/homedeliverys/manage/%homedelivery_type'] = array(
    'title' => 'View homedelivery type',
    'title callback' => 'homedelivery_type_page_title',
    'title arguments' => array(4),
    'page callback' => 'homedelivery_information',
    'page arguments' => array(4),
    'access arguments' => array('administer homedeliverys'),
    'file' => 'homedelivery.admin.inc',
  );
  $items['admin/structure/homedeliverys/manage/%homedelivery_type/view'] = array(
    'title' => 'View',
    'type' => MENU_DEFAULT_LOCAL_TASK,
  );

  $items['homedelivery/add'] = array(
    'title' => 'Add new homedelivery',
    'page callback' => 'homedelivery_add_page',
    'access arguments' => array('create homedeliverys'),
    'weight' => 1,
    'menu_name' => 'management',
    'file' => 'homedelivery.pages.inc',
  );
  foreach (homedelivery_types() as $type) {
    $type_url_str = str_replace('_', '-', $type->type);
    $items['homedelivery/add/' . $type_url_str] = array(
      'title' => $type->name,
      'title callback' => 'check_plain',
      'page callback' => 'homedelivery_add',
      'page arguments' => array(2),
      'access arguments' => array('create homedeliverys'),
      'description' => $type->description,
      'file' => 'homedelivery.pages.inc',
    );
  }
  $items['homedelivery/%homedelivery'] = array(
    'title callback' => 'homedelivery_page_title',
    'title arguments' => array(1),
    'page callback' => 'homedelivery_page_view',
    'page arguments' => array(1),
    'access arguments' => array('view homedeliverys'),
    'type' => MENU_CALLBACK,
    'file' => 'homedelivery.pages.inc',
  );
  $items['homedelivery/%homedelivery/view'] = array(
```

```
    'title' => 'View',
    'type' => MENU_DEFAULT_LOCAL_TASK,
    'weight' => -10,
  );
  $items['homedelivery/%homedelivery/edit'] = array(
    'title' => 'Edit',
    'page callback' => 'homedelivery_page_edit',
    'page arguments' => array(1),
    'access arguments' => array('update homedeliverys'),
    'weight' => 0,
    'type' => MENU_LOCAL_TASK,
    'context' => MENU_CONTEXT_PAGE | MENU_CONTEXT_INLINE,
    'file' => 'homedelivery.pages.inc',
  );
  $items['homedelivery/%homedelivery/delete'] = array(
    'title' => 'Delete',
    'page callback' => 'drupal get form',
    'page arguments' => array('homedelivery_delete_confirm', 1),
    'access arguments' => array('delete homedeliverys'),
    'weight' => 1,
    'type' => MENU_LOCAL_TASK,
    'context' => MENU_CONTEXT_INLINE,
    'file' => 'homedelivery.pages.inc',
  );
  $items['homedelivery/recent'] = array(
    'title' => 'Recently added homedeliverys',
    'page callback' => 'homedelivery_page_list_recent',
    'access arguments' => array('view homedeliverys'),
    'file' => 'homedelivery.pages.inc',
  );

  return $items;
}

/**
 * Sets the page title based on the specified homedelivery.
 *
 * @param $homedelivery
 *   The homedelivery object.
 */
function homedelivery_page_title($homedelivery) {
  return $homedelivery->title;
}

/**
 * Implements hook_forms().
 *
 * All homedelivery forms share the same form handler.
 */
function homedelivery_forms() {
  $forms = array();
  if ($types = homedelivery_types()) {
    foreach (array_keys($types) as $type) {
      $forms[$type . '_homedelivery_form']['callback'] = 'homedelivery_form';
    }
  }
  return $forms;
}

/**
 * Get a list of all homedelivery types available.
 *
 * Usually this would be more full-featured.
 *
 * @return array
 *   An array of homedelivery type objects. Note that the "type" key matches what
 *   was defined as the main key in hook_entity_info().
 */
function homedelivery_types() {
  $types = &drupal_static(__FUNCTION__);

  if (empty($types)) {
    $types['individual'] = (object)array(
      'type' => 'individual',
      'name' => t('Individual'),
      'description' => t('Individual order.')
    );
```

```
    $types['company'] = (object)array(
      'type' => 'company',
      'name' => t('Company'),
      'description' => t('Company order.')
    );
  }
  return $types;
}

/**
 * Menu title callback.
 *
 * @param $type
 *   The homedelivery type.
 * @return string
 *   The human-friendly name of the homedelivery type.
 */
function homedelivery_type_page_title($type) {
  return t('Manage @type', array('@type' => $type->name));
}

/**
 * Get an individual homedelivery type definition object.
 *
 * @param $type
 *   The key of the homedelivery type we want.
 * @return object
 *   The specified homedelivery type.
 */
function homedelivery_type_load($type) {
  $types = homedelivery_types();
  $type = str_replace('-', '_', $type);
  return isset($types[$type]) ? $types[$type] : FALSE;
}

/**
 * Load homedelivery entities from the database.
 *
 * This function should be used whenever you need to load more than one homedelivery
 * from the database. Artworks are loaded into memory and will not require
 * database access if loaded again during the same page request.
 *
 * @see entity_load()
 *
 * @param $aids
 *   An array of homedelivery IDs.
 * @param $conditions
 *   An array of conditions on the {homedelivery} table in the form 'field' => $value.
 * @param $reset
 *   Whether to reset the internal entity_load cache.
 *
 * @return
 *   An array of node objects indexed by nid.
 */
function homedelivery_load_multiple($aids = array(), $conditions = array(), $reset =
FALSE) {
  return entity_load('homedelivery', $aids, $conditions, $reset);
}

/**
 * Load an homedelivery object from the database.
 *
 * @param $aid
 *   The homedelivery ID.
 * @param $vid
 *   The revision ID.
 * @param $reset
 *   Whether to reset the homedelivery_load_multiple cache.
 *
 * @return
 *   A fully-populated node object.
 */
function homedelivery_load($aid = NULL, $vid = NULL, $reset = FALSE) {
  $aids = (isset($aid) ? array($aid) : array());
  $conditions = (isset($vid) ? array('vid' => $vid) : array());
  $homedelivery = homedelivery_load_multiple($aids, $conditions, $reset);
  return $homedelivery ? reset($homedelivery) : FALSE;
```

```
}

/**
 * Save an homedelivery.
 *
 * @param $homedelivery
 *   The homedelivery to be saved.
 * @return
 *   The saved homedelivery, now with an aid if necessary.
 */
function homedelivery_save($homedelivery) {
  return entity_get_controller('homedelivery')->save($homedelivery);
}

/**
 * Returns an initialized homedelivery object.
 *
 * @param $type
 *   The machine-readable type of the homedelivery.
 *
 * @return
 *   An homedelivery object with all default fields initialized.
 */
function homedelivery_new($type = '') {
  return entity_get_controller('homedelivery')->create($type);
}


/**
 * Deletes an homedelivery by ID.
 *
 * @param $aid
 *   The ID of the product to delete.
 *
 * @return
 *   TRUE on success, FALSE otherwise.
 */
function homedelivery_delete($aid) {
  return homedelivery_delete_multiple(array($aid));
}

/**
 * Deletes multiple homedeliverys by ID.
 *
 * @param $aids
 *   An array of homedelivery IDs to delete.
 *
 * @return
 *   TRUE on success, FALSE otherwise.
 */
function homedelivery_delete_multiple($aids) {
  return entity_get_controller('homedelivery')->delete($aids);
}
```

homedelivery.admin.inc

```
<?php

/**
 * Menu callback; List all homedelivery types available.
 */
function homedelivery_overview_types() {
  foreach (homedelivery_types() as $type => $info) {
    $type_url_str = str_replace('_', '-', $type);
    $label = t('Manage @type Orders', array('@type' => $info->name));
    $items[] = l($label, 'admin/structure/homedeliverys/manage/' . $type_url_str);
  }

  return theme('item_list', array('items' => $items));
}

/**
```

```
 * Menu callback; Artwork information page.
 *
 * @param object $homedelivery_type
 */
function homedelivery information($homedelivery type) {
  return $homedelivery_type->name . ': ' . $homedelivery_type->description;
}
```

homedelivery.controller.inc

```
<?php

/**
 * Controller for loading, creating, and saving homedeliverys.
 *
 * The default loader, which we extend, handles load() already. We only
 * need to add saving and creating.
 */
class HomedeliveryController extends DrupalDefaultEntityController {

  public function save($homedelivery) {
    $transaction = db_transaction();

    try {
      global $user;

      // Determine if we will be inserting a new homedelivery.
      $homedelivery->is_new = empty($homedelivery->aid);

      // Set the timestamp fields.
      if (empty($homedelivery->created)) {
        $homedelivery->created = REQUEST_TIME;
      }
      $homedelivery->changed = REQUEST_TIME;

      $homedelivery->revision_timestamp = REQUEST_TIME;
      $update_homedelivery = TRUE;

      // Give modules the opportunity to prepare field data for saving.
      field_attach_presave('homedelivery', $homedelivery);

      // When saving a new homedelivery revision, unset any existing $homedelivery->vid
      // to ensure a new revision will actually be created and store the old
      // revision ID in a separate property for homedelivery hook implementations.
      if (!$homedelivery->is_new && !empty($homedelivery->revision) &&
$homedelivery->vid) {
        $homedelivery->old_vid = $homedelivery->vid;
        unset($homedelivery->vid);
      }

      // If this is a new homedelivery...
      if ($homedelivery->is_new) {
        // Save the new homedelivery.
        drupal_write_record('homedelivery', $homedelivery);

        // Save the initial revision.
        $this->saveRevision($homedelivery, $user->uid);

        $op = 'insert';
      }
      else {
        // Save the updated homedelivery.
        drupal_write_record('homedelivery', $homedelivery, 'aid');

        // If a new homedelivery revision was requested, save a new record for that;
        // otherwise, update the homedelivery revision record that matches the value
        // of $homedelivery->vid.
        if (!empty($homedelivery->revision)) {
          $this->saveRevision($homedelivery, $user->uid);
        }
        else {
          $this->saveRevision($homedelivery, $user->uid, TRUE);
```

```
          $update_homedelivery = FALSE;
        }

        $op = 'update';
      }

      // If the revision ID is new or updated, save it to the homedelivery.
      if ($update_homedelivery) {
        db_update('homedelivery')
          ->fields(array('vid' => $homedelivery->vid))
          ->condition('aid', $homedelivery->aid)
          ->execute();
      }

      // Save fields.
      $function = 'field_attach_' . $op;
      $function('homedelivery', $homedelivery);

      module_invoke_all('entity_' . $op, $homedelivery, 'homedelivery');

      // Clear internal properties.
      unset($homedelivery->is_new);

      // Ignore slave server temporarily to give time for the saved order to be
      // propagated to the slave.
      db_ignore_slave();

      return $homedelivery;
    }
    catch (Exception $e) {
      $transaction->rollback();
      watchdog_exception('homedelivery', $e, NULL, WATCHDOG_ERROR);
      return FALSE;
    }
  }

  /**
   * Saves an homedelivery revision with the uid of the current user.
   *
   * @param $homedelivery
   *   The fully loaded homedelivery object.
   * @param $uid
   *   The user's uid for the current revision.
   * @param $update
   *   TRUE or FALSE indicating whether or not the existing revision should be
   *     updated instead of a new one created.
   */
  function saveRevision($homedelivery, $uid, $update = FALSE) {
    // Hold on to the homedelivery's original creator_uid but swap in the revision's
    // creator_uid for the momentary write.
    $temp_uid = $homedelivery->uid;
    $homedelivery->uid = $uid;

    // Update the existing revision if specified.
    if ($update) {
      drupal_write_record('homedelivery_revision', $homedelivery, 'vid');
    }
    else {
      // Otherwise insert a new revision. This will automatically update $homedelivery
      // to include the vid.
      drupal_write_record('homedelivery_revision', $homedelivery);
    }

    // Reset the order's creator_uid to the original value.
    $homedelivery->uid = $temp_uid;
  }

  /**
   * Deletes multiple homedeliverys by ID.
   *
   * @param $aids
   *   An array of homedelivery IDs to delete.
   * @return
   *   TRUE on success, FALSE otherwise.
   */
  public function delete($aids) {
    if (!empty($aids)) {
```

```
      $homedeliverys = $this->load($aids, array());

      $transaction = db_transaction();

      try {
        db_delete('homedelivery')
          ->condition('aid', $aids, 'IN')
          ->execute();

        db_delete('homedelivery_revision')
          ->condition('aid', $aids, 'IN')
          ->execute();

        foreach ($homedeliverys as $homedelivery_id => $homedelivery) {
          field_attach_delete('homedelivery', $homedelivery);
        }

        // Ignore slave server temporarily to give time for the
        // saved homedelivery to be propagated to the slave.
        db_ignore_slave();
      }
      catch (Exception $e) {
        $transaction->rollback();
        watchdog_exception('homedelivery', $e, NULL, WATCHDOG_ERROR);
        return FALSE;
      }

      module_invoke_all('entity_delete', $homedelivery, 'homedelivery');

      // Clear the page and block and homedelivery caches.
      cache_clear_all();
      $this->resetCache();
    }

    return TRUE;
  }

  /**
   * Create a default homedelivery.
   *
   * @param $type
   *   The machine-readable type of the homedelivery.
   *
   * @return
   *   An homedelivery object with all default fields initialized.
   */
  public function create($type = '') {
    return (object) array(
      'aid' => '',
      'type' => $type,
      'title' => '',
    );
  }
}
```

homedelivery.pages.inc

```
<?php

/**
 * Menu callback; Show list of homedelivery types we can add.
 */
function homedelivery_add_page() {
  $item = menu_get_item();
  $links = system_admin_menu_block($item);

  foreach ($links as $link) {
    $items[] = l($link['title'], $link['href'], $item['localized_options'])
      . ': ' . filter_xss_admin($link['description']);
  }

  return theme('item_list', array('items' => $items));
```

```
}

/**
 * Present an homedelivery submission form.
 */
function homedelivery_add($type) {
  global $user;

  $types = homedelivery_types();
  $type = isset($type) ? str_replace('-', '_', $type) : NULL;
  if (empty($types[$type])) {
    return MENU_NOT_FOUND;
  }

  $homedelivery = entity_get_controller('homedelivery')->create($type);

  drupal set title(t('Create @name Order', array('@name' => $types[$type]->name)),
PASS_THROUGH);
  return drupal_get_form($type . '_homedelivery_form', $homedelivery);
}

/**
 * Menu callback; presents the homedelivery editing form, or redirects to delete
confirmation.
 *
 * @param $homedelivery
 *   The homedelivery object to edit.
 */
function homedelivery_page_edit($homedelivery) {
  $types = homedelivery_types();
  drupal_set_title(t('Edit <em>@type</em> Order made in @title', array('@type' =>
$types[$homedelivery->type]->name, '@title' => $homedelivery->title)),
PASS_THROUGH);

  return drupal_get_form($homedelivery->type . '_homedelivery_form', $homedelivery);
}

/**
 * Form builder; Displays the homedelivery add/edit form.
 *
 * @param $form
 * @param $form_state
 * @param $homedelivery
 *   The homedelivery object to edit, which may be brand new.
 */
function homedelivery_form($form, &$form_state, $homedelivery) {

  // Set the id and identify this as an homedelivery edit form.
  $form['#id'] = 'homedelivery-form';

  // Save the homedelivery for later, in case we need it.
  $form['#homedelivery'] = $homedelivery;
  $form_state['homedelivery'] = $homedelivery;

  // Common fields. We don't have many.
  $form['title'] = array(
    '#type' => 'textfield',
    '#title' => t('Record Date'),
    '#default_value' => $homedelivery->title,
    '#weight' => -5,
    '#required' => TRUE,
  );

    /*
  $form['revision'] = array(
    '#access' => user_access('administer homedeliverys'),
    '#type' => 'checkbox',
    '#title' => t('Create new revision'),
    '#default_value' => 0,
  );
    */

  // Add the buttons.
  $form['buttons'] = array();
  $form['buttons']['#weight'] = 100;
  $form['buttons']['submit'] = array(
    '#type' => 'submit',
```

```
    '#value' => t('Save'),
    '#weight' => 5,
    '#submit' => array('homedelivery_form_submit'),
  );
  if (!empty($homedelivery->aid)) {
    $form['buttons']['delete'] = array(
      '#access' => user_access('delete homedeliverys'),
      '#type' => 'submit',
      '#value' => t('Delete'),
      '#weight' => 15,
      '#submit' => array('homedelivery_form_delete_submit'),
    );
  }

  $form['#validate'][] = 'homedelivery_form_validate';

  field_attach_form('homedelivery', $homedelivery, $form, $form_state);

  return $form;
}

function homedelivery_form_validate($form, &$form_state) {
  $homedelivery = $form_state['homedelivery'];

  // Field validation.
  field_attach_form_validate('homedelivery', $homedelivery, $form, $form_state);
}


function homedelivery_form_submit($form, &$form_state) {
  global $user;

  $homedelivery = &$form_state['homedelivery'];

  // Set the homedelivery's uid if it's being created at this time.
  if (empty($homedelivery->uid)) {
    $homedelivery->uid = $user->uid;
  }

  $homedelivery->title = $form_state['values']['title'];
  //$homedelivery->revision = $form_state['values']['revision'];

  // Notify field widgets.
  field_attach_submit('homedelivery', $homedelivery, $form, $form_state);

  // Save the homedelivery.
  homedelivery_save($homedelivery);

  // Notify the user.
  drupal_set_message(t('Home Delivery saved.'));

  $form_state['redirect'] = 'homedelivery/' . $homedelivery->aid;
}


function homedelivery_form_delete_submit($form, &$form_state) {
  $destination = array();
  if (isset($_GET['destination'])) {
    $destination = drupal_get_destination();
    unset($_GET['destination']);
  }
  $homedelivery = $form['#homedelivery'];
  $form state['redirect'] = array('homedelivery/' . $homedelivery->aid . '/delete',
array('query' => $destination));
}

/**
 * Displays an homedelivery.
 *
 * @param $homedelivery
 *   The homedelivery object to display.
 * @param $view_mode
 *   The view mode we want to display.
 */
function homedelivery_page_view($homedelivery, $view_mode = 'full') {
  // Remove previously built content, if exists.
  $homedelivery->content = array();
```

```
  if ($view_mode == 'teaser') {
    $homedelivery->content['title'] = array(
      '#markup' => filter_xss($homedelivery->title),
      '#weight' => -5,
    );
  }

  // Build fields content.
  field_attach_prepare_view('homedelivery', array($homedelivery->aid =>
$homedelivery), $view_mode);
  entity_prepare_view('homedelivery', array($homedelivery->aid => $homedelivery));
  $homedelivery->content += field_attach_view('homedelivery', $homedelivery,
$view_mode);

  return $homedelivery->content;
}

/**
 * Form bulder; Asks for confirmation of homedelivery deletion.
 */
function homedelivery_delete_confirm($form, &$form_state, $homedelivery) {
  $form['#homedelivery'] = $homedelivery;
  // Always provide entity id in the same form key as in the entity edit form.
  $form['aid'] = array('#type' => 'value', '#value' => $homedelivery->aid);
  return confirm_form($form,
    t('Are you sure you want to delete this order made in %title?', array('%title' =>
$homedelivery->title)),
    'homedelivery/' . $homedelivery->aid,
    t('This action cannot be undone.'),
    t('Delete'),
    t('Cancel')
  );
}

/**
 * Executes homedelivery deletion.
 */
function homedelivery_delete_confirm_submit($form, &$form_state) {
  if ($form_state['values']['confirm']) {
    $homedelivery = homedelivery_load($form_state['values']['aid']);
    homedelivery_delete($form_state['values']['aid']);
    watchdog('homedelivery', '@type: deleted %title.', array('@type' =>
$homedelivery->type, '%title' => $homedelivery->title));

    $types = homedelivery_types();
    drupal_set_message(t('@type %title has been deleted.', array('@type' =>
$types[$homedelivery->type]->name, '%title' => $homedelivery->title)));
  }

  $form_state['redirect'] = '<front>';
}

/**
 * Menu callback; Displays a listing of recent homedeliverys.
 *
 * This doesn't really work yet because our presentation code doesn't show
 * the title.
 */
function homedelivery_page_list_recent() {
    global $user;

    //Added part for render header rows in a table.
    $header_homedelivery = array(
        array('data' => t('Type')),
        array('data' => t('Record Date')),
        //array('data' => t('Record Time')),
        array('data' => t('Customer Name')),
        array('data' => t('Customer Address')),
        array('data' => t('Phone Number')),
        array('data' => t('Full Order Info')),
        array('data' => t('Record Person')),
    );

  //$content = array();

  $query = new EntityFieldQuery();
```

```
  $query
    ->entityCondition('entity_type', 'homedelivery')
    ->propertyOrderBy('title', 'ASC')
    //->fieldCondition('field_cname', 'value', '', 'CONTAINS', 0)
    ->range(0, 5);
  $result = $query->execute();

  $homedeliverys = homedelivery_load_multiple(array_keys($result['homedelivery']));
  foreach ($homedeliverys as $homedelivery) {
    //The following line is not needed anymore since we will present whole data as a
table.
    //$content[$homedelivery->aid] = homedelivery_page_view($homedelivery,
'teaser');
    //instead we will define all values to be presented as $rows_homedelivery[].
    $rows_homedelivery [] = array(
        array('data' => $homedelivery->type),
        array('data' => $homedelivery->title),
        array('data' => $homedelivery->field_cname['und'][0]['value']),
        array('data' => $homedelivery->field_caddress['und'][0]['value']),
        array('data' => $homedelivery->field_cnumber['und'][0]['value']),
        array('data' => l(t('View'), 'homedelivery/' . $homedelivery -> aid)),
        array('data' => l($user->name, 'user/' . $user -> uid)),
    );
  }
  //$caption_homedelivery = t('Table for home deliveries');
  //And instead of return $content
  //return $content;
  //We will implement theme() as return value.
  $build['homedelivery/recent'] =   array(
    '#theme' => 'table',
        '#header' => $header_homedelivery,
        '#rows' => $rows_homedelivery,
        //'caption' => $caption_homedelivery
  );

  return $build;
}
```

APPENDIX 2: Codes of Module for Customer Information

customer.info

```
name = Customer Information
description = Display Customer information recorded in Homedeliveries.
core = 7.x
package = HDCRM
files[] = customer.module
dependencies[] = homedelivery
```

customer.module

```
<?php
/**
 * Implements hook_menu().
 */
function customer_menu() {
    $items['customer/all'] = array(
    'title' => 'Customer Information',
    //'type' => MENU_NORMAL_ITEM,
    'page callback' => 'drupal_get_form',
    'page arguments' => array('customer_phone_unique_autocomplete'),
    'access arguments' => array('view homedeliverys'),
    //'file' => 'ajax_example_autocomplete.inc',
    'weight' => 11,
  );
  $items['customer/all/unique_node_autocomplete_callback'] = array(
    'page callback' => 'customer_phone_unique_node_autocomplete_callback',
    //'file' => 'ajax_example_autocomplete.inc',
    'type' => MENU_CALLBACK,
    'access arguments' => array('view homedeliverys'),
  );

  return $items;
}

/**
 * A unique autocomplete form which looks up customer information by phone number in
the homedelivery table, but must keep track of the aid, because
 * phone number are certainly not guaranteed to be unique.
 *
 * @param $form
 * @param $form_state
 * @return array
 */
function customer_phone_unique_autocomplete($form, &$form_state) {

  $form['info'] = array(
    '#markup' => '<div>' . t("Enter customer phone number to searching existing customer
information.") . '</div>',
  );

  $form['phonenumber'] = array(
    '#type' => 'textfield',
    '#phonenumber' => t('Enter phone number here:'),
    // The autocomplete path is provided in hook_menu in ajax_example.module.
    '#autocomplete_path' => 'customer/all/unique_node_autocomplete_callback',
  );

  $form['actions'] = array(
    '#type' => 'actions'
  );

  $form['actions']['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Submit'),
  );
```

```
  return $form;
}

/**
 * Submit handler for node lookup unique autocomplete.
 *
 * This funtion does not work well at this moment,
 * thus, submitting triggers nothing other than set a message saying phone number is
found.
 * @param $form
 * @param $form_state
 */
function customer_phone_unique_autocomplete_submit($form, &$form_state) {
    drupal_set_message(t('Phone number ' . $form_state['values']['phonenumber'] . '
is found.'));

    //$phonenumber = homedelivery_load($form_state['values']['phonenumber']);
  //drupal_set_message(t('You found homedelivery with phone number %phonenumber',
array('%phonenumber' => $phonenumber->field_cnumber_value)));
}

/**
 * Autocomplete callback for customer information by phone number.
 *
 * @param $string
 *   The string that will be searched.
 */
function customer_phone_unique_node_autocomplete_callback($string = "") {
  $matches = array();
  if ($string) {
    $result = db_select('field_data_field_cnumber')
      ->fields('field_data_field_cnumber', array('entity_id',
'field_cnumber_value'))
      ->condition('field_cnumber_value', db_like($string) . '%', 'LIKE')
      ->range(0, 10)
      ->execute();
    foreach ($result as $phonenumber) {
      $matches[$phonenumber->field_cnumber_value] =
check_plain($phonenumber->field_cnumber_value);
    }
  }

  drupal_json_output($matches);
}
```